Table of Contents

1 INTRODUCTION

1.1 INTRODUCTION

Survent is a computer-aided interviewing software package. Interviews are conducted on personal computers — which may be connected by a network for supervisor, quota control and phone management — or on terminals on larger systems.

Survent is designed for advanced telephone interviewing, sample and quota management, sound recording, and for interviewing in shopping malls, stores, or at conventions. It is a sophisticated and flexible software program that can be used for a variety of research functions. In this manual, the name Survent is used when referring to the overall package. Individual program names are used for specific references to programs (i.e., Survent or PREPARE).

1.1.1 How Survent works

Using Quick Survent or a text editor and the Survent or Mentor PREPARE module, the spec writer is able to format and compile the questionnaire to exact specifications without extensive experience. The compiled questionnaire is tested for details, such as logic conditions and screen formatting. If you are using a shared files system, all data and quota controls are centrally stored and accessed.

There is a quota management and phone sample management control system you can use with Survent. See *Chapter 6: Telephone Number Management System*.

Supervised interviewing is designed for multiple interviewing stations with interview controls (e.g., what station runs which interview, which interviewer has reached quota limit, etc.) at the Supervisor's station.

Additional options include sound servers, Predictive dialers and autodialers, and webSurvent for surveys over the Internet.

1.1.2 How to use this manual

This manual is organized to help you understand the Survent process, from the initial setting up of the questionnaire, to testing questionnaire specs through actual interviewing.

It is a detailed guide to the specifics required for successful questionnaire design and interviewing.This is not a step-by-step introductory guide. It is a reference tool to be used by both beginners and advanced users. Even if you are completely familiar with the program, you can refer to the detailed index to quickly locate specific information.

Many examples are included. Because of the importance of these details and their spacing, the examples are set in a different font:

Example:

```
EX:

examples are indented and set in this font
```

Required parts of entries are in uppercase; variable or optional syntax is in lowercase. Bold type indicates a required entry using those exact characters; underlined text requires a user-supplied entry. Many keywords can be abbreviated. For consistency, we have used the full keyword in most places in the chapters. Exceptions to this are question types, which often work only using the shorthand version. See Appendix B for a list of allowed abbreviations.

```
 EX:

{!ROTATE,type,number in block,label}
```

In the example above, {!ROTATE} is the only required syntax. Type, number in block, and label are optional (not underlined). These three entries are not entered literally. A letter is entered for type; a number for number in block; and a question label for label. Commas are used as dividers; if they are required as placeholders, you will be warned.

```
 Syntax:

>CREATE_DB dbfilename,option
```

In the previous example, >CREATE_DB is the required meta command and dbfilename is the required user-supplied entry; option is not part of the required syntax.

When displaying the screen along with the syntax and a possible response, the example will look like this:

EX: SPEC FILE --> BANK.SPX

In the previous example, you are telling PREPARE the name of the specification file BANK.SPX.

For most commands, a Return (or enter) is required after the entry—it is not always shown; when it is shown, it is **Enter**.

The control key is indicated by **Ctrl**; the escape key by **ESC**. Functions are shown as **F1**, **F2**, etc.

1.1.3 Software bugs

The dynamic nature of CfMC software may lead to situations where there may be instances of bugs and/or differences across platforms. We try to document and/or remedy bugs as soon as we know about them — your promptness in reporting the specifics of any bugs you find can ensure that the correction of bugs takes place in a timely process.

1.1.4 Customer Support

For assistance with a problem you cannot solve, email or call CfMC Support:

E-mail: support@cfmc.com

Phone: (415) 777-2922

Fax: (415) 777-3128

(6 a.m. to 5 p.m., Pacific Time, Monday through Friday, excluding holidays)

Voicemail will record your message at all other times.

Password Protected Support files: http://distrib.cfmc.com/support or e-mail getfiles@cfmc.com.

WHERE DO I GO FROM HERE?

We hope you find this manual easy to use. CfMC is always looking for feedback regarding our software, our manuals and our technical support. Below is a list of ways to contact us, depending on what your needs are.

If you want to:	Contact	Telephone	E-mail or URL
Ask a general question	Receptionist	(415) 777-0470	
Lease CfMC software, add more users or get an upgrade	East region	(212) 777-5120	joycer@cfmc.com
	West region	(415) 777-0470	sales@cfmc.com
	Europe	011-44-20-7837-5214	shughes@cfmc.com
	Marketing		
Get help using the software. Have a suggestion for the software. Want a new feature added	Tech support	(415) 777-2922	support@cfmc.com
Get training	Training	(415) 777-0470	train@cfmc.com
Documentation: Get additional copies of manuals; report errors in manuals, documentation	Documentation	(415) 777-0470	doc@cfmc.com
Get the latest CfMC news CfMC Topline (electronic newsletter)	CfMC Web site	(415) 777-0470	www.cfmc.com
Talk to other CfMC users	Spec-talk discussion group		spec-talk@cfmc.com

Pick up files from CfMC, Distribution site (password from Support), Send files to CfMC	Bulletin Board System Software distribution CfMC FTP site	(415) 896-2362 (modem)	bbs@cfmc.com http://distrib.cfmc.com ftp://ftp.cfmc.com
Get a quote from CfMC Service Bureau or CfMC's Web site	San Francisco office	(415) 777-0470	sf_sb_mgr@cfmc.com
Get quote from Denver Service Bureau or CfMC-Denver Web site	Denver office	(303) 860-1811	denver@cfmc.com www.cfmc.com/denver
Get a quote from the New York Service Bureau	New York office	(212) 777-5120	sf_sb_mgr@cfmc.com

1.1.5 Operating System Differences

Survent works on multiple platforms. Currently, IBM-PC compatible machines running under DOS, HPUX, and PCs or terminals running SCO (PC), LINUX (PC; Red Hat 5.2+), AIX (RS6000) or SUN (Solaris 2.6+ on a Sparc chip) UNIX are supported. webSurvent works on any machine with a browser. Other machines and operating systems may be supported in the future.

HARDWARE DIFFERENCES

There are some hardware differences between PCs and terminals. In DOS, only PCs may be run; in UNIX, terminals or PCs with a terminal emulator. On a PC, you can use the HOME/END/Pg Up/Pg down and arrow keys to move around, while on a terminal, you neeed to use the CTRL key along with G,V,T and E (Home, Pg Up, Pg Down, End) and U,D,R,L (for up, down, right and left). On PCs, you get "graphic" characters displayed for boxes, etc. while on terminals you get only a nongraphic box. Colors, bold, inverse, and flashing work on all PCs but only some terminals. On PCs, all extended language characters will work. On a terminal, only some, or only after changing the system configuration will they work. In webSurvent, you get the full functionality of whatever browser you are using.

FILE NAMING CONVENTIONS

DOS and UNIX require slightly different commands and file naming conventions. Where such differences exist, they will be noted (see specific references to DOS or UNIX). We do not specifically differentiate between DOS’s \directory\subdirectory and UNIX’s /directory/subdirectory references, although obviously users must enter the syntax specific to their operating system.

Filenames can be specified in upper or lower case in DOS. In Unix, by default filenames must be lower case if specified outside of CfMC programs, but within CfMC programs the filename can be upper or lower case and we will convert them to lower case before looking for them. The files themselves and directories they reside in must be lowercase (unless >-casesensitve is set) or CfMC programs will not find them.

NOTE: This manual is written with the prerequisite that you have a SET PATH=\CFMC\GO path assignment (in DOS), or “setenv PATH /cfmc/go” command in UNIX. Program calls will be shown in shortened formats; for example, MENTOR, rather than \CFMC\GO\MENTOR. If a path is not set, you will need to give complete path information.

1.2 FILE MANAGEMENT

You should familiarize yourself with the operating system commands necessary for proper file management. If the operating system is new to you, refer to your operating system manual for complete information.

You should also be familiar with the use of an editor or word processing package. You can often edit files with a Windows-based text processor even if the file server is a UNIX machine.

1.2.1 Survent Files

DOS accepts file names of up to eight characters only by default. In addition to any operating system rules for file names, you are also restricted to file names starting with a letter and consisting only of letters, numbers and underscores (_). File names are followed by three-character extensions, or suffixes, separated from the file name by a period. Survent, like most microcomputer programs, provides default extensions for files it creates to help identify a file's type and purpose. Extensions are part of the file name and should be specified when renaming or copying files using DOS or UNIX commands.

Directories referred to have the same limitations; if you refer to files in a directory with a long name, you may get errors. In UNIX, you can override the filename and directory limitations by putting the filename in quotes ("name"). In DOS, you can use the short filenames generated by the system in conjunction with the long filename otherwise given (see DIR command), but you cannot use the long name.

Here is a list of files provided by CfMC, or created by the CfMC programs, where they reside on the computer, and a short description of their use. Most of these will be discussed in later chapters. See *Appendix G* used by the phone system. See the *Utilities* manual for files related to the utilities and the Mentor manual for files related to that program.

PROGRAM FILES

Program files control all Survent-related functions. You should have a path set to the \CFMC\GO directory to access the CfMC programs and BAT files. DOS commands run files with .EXE extension. UNIX commands run lowercase filenames with no extension.

NAME	FUNCTION
FIXRESUME	Verifies that suspended interviews can be resumed if the questionnaire is modified
FONEBULD	Creates compiled phone file from ASCII file

FONESIM	Simulation program for phone system
FONEUTIL	Displays and modifies phone file parameters
MAKECFG	Updates study and interviewer information file
MAKEMSG	Add custom messages to CfMC msgfile
MAKEPREP	Makes hardcopy text into questionnaire specifications
MENTOR	Creates questionnaires, copies and converts data files, and does data manipulation operations
QUOTAMOD	Displays/modifies quota values from interview(s)
RAWCOPY	Program that recovers corrupt data files
STUDYSRVR	CfMC server, started up typically by SERVER.BAT
SURVENT	Survent interviewing program
SURVMON	Allows monitoring of interviews
SURVSUPR	Interview supervisor program started with “SUPER” command
SUSPRES	Displays responses in a suspended interview file

SYSTEM FILES

System files are used to hold information for the program files, the are kept in the “control” directory. They must be available for the programs to run properly.

msg####	File of messages displayed by the programs. These include error messages, prompts, informational items, and help screens. If you have the wrong version of the message file, or the program cannot find it, the program will not run.
parmfile	File with the software validation string and other parameters needed by the CfMC server. This must be present for Survent to operate.
ttyinfofile	File with description of the stations, must be present but can be blank if using system variables.
zonetable	File with description of the stations, must be present but can be blank if using system variables.

There are also optional files initial and mentinit which the programs will read to set parameters before operating. You can customize these control files and strategically locate them. The search order theprograms use when looking for these files is:

1 The current logon directory

2 The CONTROL subdirectory in logon directory

3 The \CFMC\CONTROL subdirectory

COMMAND FILES

Command files call the programs with specific controls. You should have a path set to \CFMC\GO to access these command files. DOS names have a .BAT extension, UNIX names must be in lower case. DOS .BAT files run program names that end in .EXE; Unix command files run lower-case named programs.

MONITOR	Runs SURVMON
MENTOR or PREPARE (DOS)	Runs Mentor to compile a questionnaire reformat
REFORMAT	Reformats data files from compressed binary to ASCII; spreading multi-punch data. Also makes a map of the data
SERVER	Runs stdysrvr
START	Runs Survent
SUPER	Runs survsupr

PREPARE FILES

"Prepare" files are created by Mentor's ~Prepare module when it is run. These files are created in the current local directory (wherever you are signed on). See *2.2.1 FILES CREATED BY PREPARE* for more information on these files (and others not listed here).

studyname.chk	Lists compiled questions and their data locations.
studyname.db	Data base file storing v ariables for Mentor and Utilities.
studyname.hrd	Hardcopy listing produced when HARD_COPY compiler command is used.
studyname.qff	Questionnaire file.
studyname.qsp	ASCII backup spec file.
studyname.quo	Qupta/ID file
studyname.sum	Lists question number, data location and first line of text.

SURVENT FILES

Survent files are created by the Survent program.

resumename	Resume file, stores data up to the point where the interview was SUSPENDed. This file is located in the subdirectory studyname.R_ (DOS, UNIX), and is named by the user in standalone mode.
studyname.tr	Data file with data collected during interviews. In standalone mode, created in local directory. In shared files mode, created in |$HOME\DATA

UTILITY FILES

Utility files are created by Survent utilities in the local directory.

studyname.def	Program language conversion file for REFORMAT.
dataname.lst	Listing of variable values from LIST program..
studyname.rfl	Map of data locations, created by REFORMAT.
studyname.rft	Spread ASCII data, created by REFORMAT.
dataname.scn	Summary tables created by SCAN utility.

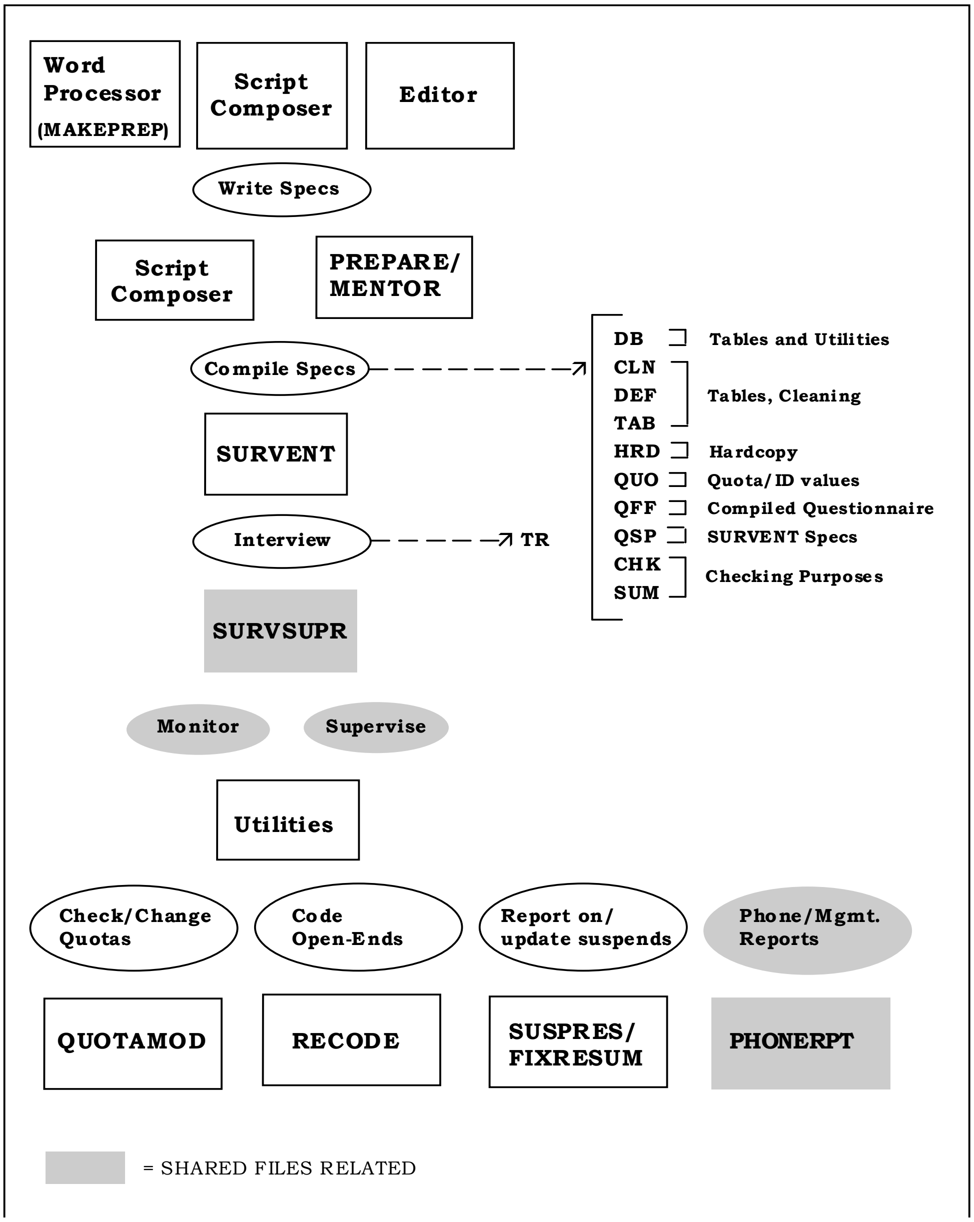
You must know which directory a file is in before giving a command. The default is your current drive and directory. When in doubt, use the ls (Unix) or DIR(DOS) command to review the files on a drive (e.g., DIR C:\command).

1.3 PICTORIAL OVERVIEW

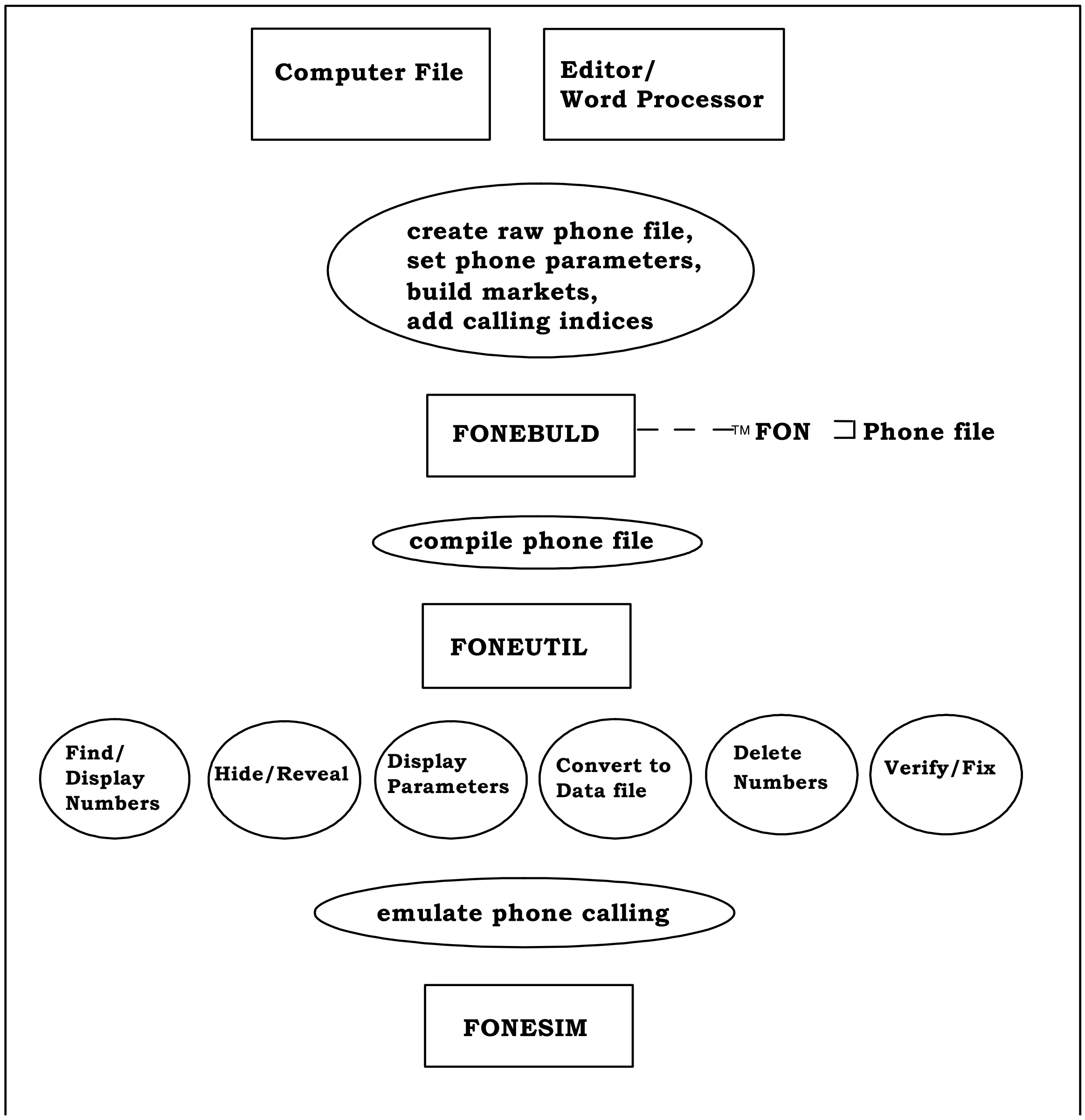
SIMPLE STEPS FOR STANDALONE, SUPERVISED MODES

STANDALONE	**SUPERVISED**
QUESTIONNAIRE	DESIGN QUESTIONNAIRE
COMPILE SPECIFICATIONS	COMPILE SPECIFICATIONS
TEST	
	COMPILE PHONE FILE
	TEST
INTERVIEW	INTERVIEW
	SUPERVISE/MONITOR
RUN UTILITIES	RUN UTILITIES

***Programs*, *Processes* and Files**



Phone System



2 GETTING STARTED

2.1 INTRODUCTION TO PREPARE

The PREPARE module of Mentor is used to create questionnaires for CfMC's Survent interviewing program. In writing questionnaires for PREPARE, you will create the question text, formatting commands, logic conditions and other program commands which will be used by Survent. If you already have a word processor version of the questionnaire, convert it to ASCII and you can use the MAKEPREP utility to convert the text into specifications for you. See *5.4 MAKEPREP* for more details. There are several ways to create PREPARE specifications:

The Script Composer is a menu-driven, interactive way of creating specifications. With the fullscreen icon editor, user-friendly menus and help screens, you can create specifications without learning syntax that would otherwise required. Many special features are available. The *Script Composer* manual provides step-by-step instructions for creating questionnaires in this mode. This is an add-on product available with a separate license.

Specification mode allows you to write specifications and program commands line-by-line, either interactively or in a file to be read by the program in batch mode. PREPARE checks the syntax of each line as it is read, and produces error or warning messages where required. PREPARE's interactive and batch specification modes require the same command and specification syntax.

In interactive mode, you enter the specifications and program commands line-by-line while working within the PREPARE module.

In batch mode, you write specifications in an ASCII text file and read it into the PREPARE module for processing. This allows you to modify and recompile without retyping.

You may also switch between Script Composer and specification modes.

- When working in the Script Composer, compile creating a QSP file, and use that file as your specification file in PREPARE.
- Read a specification file into the Script Composer at the startup screen and make further changes in the Script Composer.

This chapter covers general PREPARE information, creating and using specification files, and specific syntax for writing commands and question specifications. The different question types and their uses are explained, as well as other program control commands.

This information will be presented in the following order:

1. General introduction to PREPARE
2. Files used in PREPARE
3. Required elements of the questions (question label line and question type)
4. Optional elements (question text and condition statements)

2.2 USING PREPARE

To access the PREPARE subsystem at the operating system prompt, enter the following:

PREPARE (DOS)

MENTOR (UNIX)

DOS NOTE: If you get a message *bad command or filename* after typing PREPARE, you do not have a path set pointing to the CfMC programs. Usually, you can do this by typing SET PATH=\CFMC\GO.

You should have a path set, as well as other configuration parameters, to run CfMC software. If you do not, you should contact the person in charge of the installation, re-install the Survent software package, or call CfMC Customer Services. PREPARE is a batch (.BAT) file which executes the Mentor program.

When you type the access command, the program will prompt you for a possible specification (spec)

file:

```
     Spec file -->
```

The spec file is the file containing questionnaire specifications and other commands. It must be an ASCII file. It must NOT contain tabs or other format control characters often used by word processors. The program reads the file line by line.

If you have a file, you would enter the file name, or the drive and directory (or group and account) of the file, if they differ from your current drive and/or directory, with the file name. The only required element is the file name.

The syntax for the file name is:

```
Drive:\Directory\SubDirectory\filename.extension (DOS)
Directory/Subdirectory/filename.extension (UNIX)
```

The drive letter, if needed, is first, followed by a colon (:) (DOS only). The default is the drive you are currently signed on to. If the file is located in a directory and/or sub-directory other than the one you are signed on to, enter the directories separated by slashes (\ for DOS; / for UNIX). This is followed by the file name, which must be from one to eight alphanumeric characters long, starting with a letter. We suggest you use an extension on the file name, to specify the type of file it is. The file name extension can be from one to three alphanumeric characters, preceded by a period (.).

```
     EX: C: \ROGER\ACME\RX123.QPX
```

The extension (suffix) is a naming convention that is left to the user. It can be used to identify 'types' of files. For example, you could use QPX or SPX for your specification files. Be careful not to use standard DOS or CfMC extensions to identify your files. (See *1.2 FILE MANAGEMENT* and *Appendix G: HARDWARE AND SYSTEM CHECKLIST* for a list of the CfMC files and extensions.) You cannot use a file with the extension of QSP because that us used for CfMC backup spec files.

The maximum filename extension is 20 characters. That way, dates can be used in the entensions (using a YYMMDDHHMMSS format).

The spec file may include references to other files, bringing in separate portions of questionnaire specifications. (Using file references is explained in *3.3.2 FILE ACCESS COMMANDS.)*

If you want to enter commands line by line, press **Enter** at this prompt to tell the program that the input for this questionnaire will come from your keyboard. Otherwise, type the name of the file with the commands needed to create your questionnaire.

The program will then prompt you for a list file:

```
    List file -->
```

The list file is where the list of program commands processed and resulting messages go. Press **ENTER** to tell PREPARE that you wish to see these at your screen. Specifying PRN as the list file will send the output to the printer on most DOS machines; LP or LJ works on most HP systems; there is no easy way to send the output directly to the printer in UNIX. You may also specify a file name and the output will be stored in a disk file of that name. If the name is preceded by a dash, a same named file will be purged. (***NOTE:*** >PURGE_SAME does not affect the list file. See *3.3.2 FILE ACCESS COMMANDS* for more information.)

The spec file and list file may also be specified as part of the command line:

```
    EX: PREPARE BANK.QPX -BANK.LFL       (DOS)
    MENTOR BANK.QPX -BANK.LFL            (UNIX)
```

Because the spec file (BANK.QPX) and the list file (BANK.LFL) are specified, you would not be prompted for them. The program would process the commands from BANK.QPX and display the commands processed and resulting program messages in the file BANK.LFL. If BANK.LFL already exists, it will be purged and replaced with the new version because of the dash preceding the name.

The keyboard and/or console (computer screen) can also be specified as the spec and/or the list file on the command line. *CON* is the file name used for this type of standard input and output:

```
EX: PREPARE CON CON      (DOS)
MENTOR CON CON           (UNIX)
```

If you give a file name as the spec file, PREPARE expects to read that file only (and any references within that file) and then exit. If you want to read a file to *start* a questionnaire building session and then continue interactively from the end of that file reference on, you can use an ampersand (&) prior to the spec file name, or "in:-1" in UNIX (in:-3 replaces &&& in UNIX).

```
EX: PREPARE "&BANK.SPX" "CON" (DOS)
 mentor "&bank.spx" "con"     (UNIX)
```

In the above example the file would be read and processed to the end, then the program would prompt for more commands.

USING SYSTEM VARIABLES IN THE FILE NAME

You can set a variable at the operating system level, (for example, in DOS, *SET DRIVE=C; in UNIX, setenv name,* and reference it wherever you can reference a file name in CfMC programs. This would be at the spec or list file prompt, or when using &filename references. To reference the variable *DRIVE* above, put exclamation points (!) around it at the start of the file name reference. For example, *!DRIVE!Myfile.LST* would look for C:Myfile.LST.

The variable named *CFMC* has a special meaning. Do not use it for other uses. Wherever you set it is where the program(s) look for CfMC files. The default for the variable CFMC is \CFMC\.

COMMAND LINE PARAMETERS

To run with more memory (faster):

```
MENTOR "CON" "CON" "CORE:4000000"
```

To load a predefined variable from the environment:

```
MENTOR "CON" "CON" "DEFINE:@FILE=%FILENAME%" (DOS)
MENTOR "CON" "CON" "DEFINE:@FILE=$FILENAME" (UNIX)
```

To preload things into a program such as Survent (study name, ID, etc.) use the INIT: parameter:

```
    SURVENT CON CON “INIT:<item1;item2…>”\
```

2.2.1 Files created by PREPARE

The PREPARE module produces some files automatically and some when you request them. When you begin a new questionnaire, you specify a study name in the header statement that PREPARE will use to name the files it creates. The file names take the form of your study name plus a suffix or extension. Here are some of the files created and their suffixes:

File type	**Suffix**	**File name Study name: DEMO**
Compiled questionnaire	QFF	DEMO.QFF
Quota/Case ID	QUO	DEMO.QUO
- Variables data base	DB	DEMO.DB
-Backup Qnaire spec file	QSP	DEMO.QSP
-Ordered data list	CHK	DEMO.CHK
-Summary question list	SUM	DEMO.SUM
*Hardcopy question list	HRD	DEMO.HRD
*Cleaning specifications	CLN	DEMO.CLN
*Table definitions	DEF	DEMO.DEF

- indicates a file you may suppress if you don’t want it.

***** indicates a file you’ll get if you request it.

NOTE:

- If a question has no label, information about it will be stored using the program-assigned label preceded by CFM (i.e., CFM0023).
- If the questionnaire has errors, only the QSP file will be created

- Files not noted below as ASCII are not directly readable (i.e., by an editor); they are meant to be accessed by a CFMC program.

COMPILED QUESTIONNAIRE FILE AND QUOTA FILE (QFF)

PREPARE creates a compiled questionnaire and quota file to be used for the actual interview with the Survent program. The QFF file contains all the screen text, formatting commands and logic conditions needed to collect interview data. If necessary, you can generate any of the other informational files from the QFF file. (See the end of *2.3.1 COMPOSING AND COMPILING QUESTIONS IN PREPARE* for more details.)

QUOTA FILE (QUO)

PREPARE creates a file to keep track of quota values and counters even if no QUOTA statements or functions are present in your specs. This file can be accessed using the QUOTAMOD utility and the Supervisor's QSS command, and can be suppressed using the -QUOTA_FILE header option. Supervised and shared-file Survent require a quota file. Standalone Survent can be run without a quota file.

VARIABLES DATA BASE FILE (DB)

The DB file is created to store compiled variables for Mentor and the utilities to use. You will get a DB file unless you specify -DB_FILE on your header statement or compile with ~PREPARE COMPILE -SPEC_FILES. PREPARE will create the file using default specifications unless you have a CREATE_DB or DB_SIZES meta command before your header statement. PREPARE and Survent do not use the DB file.

BACKUP QUESTIONNAIRE SPECIFICATION FILE (QSP)

PREPARE creates a backup specification file when a questionnaire is compiled. This is an ASCII record of all the questionnaire specifications. You can edit it with a text editor and reprocess it by typing its name at the spec file prompt, or load it into SCRIPT COMPOSER to continue modifying your specifications. The specification file, or portions of it, can also be accessed for creating new questions. If you wish to edit this file and use it as input to PREPARE, you must rename it with an extension other than QSP.

This backup file will have all repeats expanded so that each question is displayed individually. A comment is generated with the question number and data location for each question. Comments (text after leading 's or ' 's) and {!COMMENT} blocks are be retained.

The file will have data locations hard coded if you used the HARD_CODE compiler directive. You may then edit this file to make additional changes and guarantee that locations you let the program assign do not get moved when adding or removing questions (See *2.9.1 WHAT TO DO WHEN A STUDY HAS BEEN SOFT CODED*).

ORDERED DATA LIST FILE (CHK)

PREPARE prints a map of column locations used, corresponding question numbers with types, and other uses of the location. This lets you check to make sure the same column was not inappropriately assigned to more than one question. This map is sorted in column order. This file can be suppressed using the -CHECK_FILE header option.

The CHK file lists:

- Case ID field
- Record ID and content fields (if CARD_FORMAT is used)
- Data column location and width
- Question label
- Question number
- Question type and subtype
- Other use of same data location
- Related labels referred to
- Gaps of columns
- Total case length

- Text area starting location
- Start of loop iterations

NOTE: All data modifying questions are listed. Any question type with a subtype A will not be listed because that question does not modify data. PHO,G and PHO,P statements that show where data is referenced in the phone file will also be listed.

```
EX: RUNR.CHK
Data area ends at 3/80, Work area starts at 4/1,
Text area starts at 5/1
Col.wid Label QQ# Qtype Sub Other use LabelRef
1/6.1 QN1 0.10 [CAT ]
1/7.1 QN2A 0.20 [CAT ]
1/8.1 QN2B 0.40 [CAT ]
1/9.1 QN2C 0.50 [CAT ]
1/10.1 QN2D 0.60 [CAT ]
1/11.1 QN3 1.10 [CAT ]
1/12.2 QN4 1.20 [NUM ]
GAP of 7 columns
1/20.1 QN5A 1.30 [CAT ]
1/21.1 QN5B 1.40 [CAT ]
1/22.1 QN5C 1.50 [CAT ]
GAP of 35 columns
1/57.1 QN21 4.40 [CAT ]
1/58.2 LAST 4.70 [FLD ] N
4/71.1 DEMO2 4.60 [TEXT]
Caseid at 1/1.4
Text starts at 5/1
```

NOTE: The meta command >LOCATION_FORMAT will let you specify how the data locations print out (absolute column or 80 column record #/column).

Summary Question List File (SUM)

PREPARE creates an ASCII question list file that can be useful for checking the contents of a questionnaire file. This is a simplified listing which contains a one-line description for each question in question order. This file can be suppressed using the -SUMMARY_FILE header option.

Each line of the SUM file lists:

- Label
- Question number
- Data column location and width
- Question type and subtype
- The first n characters of the question text. N is determined by the SPEC_WIDTH setting on your study header (i.e., the text starts in column 39 so if your SPEC_WIDTH is 132, you'll get up to 93 characters of text. This file will print up to position 80 by default and may be printed up to 5,000 characters wide, depending on the SPEC_WIDTH setting.

```
EX:
Label Qnum Col Wid Type Sub Text
Card 2.00 1/6 1 CAT Q2A. Which of the following ...
```

It will also list columns used by ROTATE questions and the beginning and end of SUSPEND, RESUME, ROTATE and SPECIAL blocks.

HARDCOPY QUESTION LISTING FILE (HRD)

PREPARE creates a hardcopy listing of the questionnaire when you use the compiler command HARD_COPY. You can use various options to control how the hardcopy is presented. If you have multiple languages in the questionnaire, you may specify which language to use in the hardcopy listing. This file could be used to do paper and pencil data collection or as an easy-to-read reference for a client. Syntax is converted to readable language wherever possible.

Following is an example:

```
September 17, 2004 - Exam1 - Example Questionnaire - Page 1
__________________________________________________________
      Question: CNTRY (MAX 2 RESPONSES)
      (50.4) 01 ............ United Kingdom
      02 ............ France
      03 ............ Germany
      04 ............ Spain
```

In addition, the .HRD file is made automatically each time you compile.

The "hardcopy" file is an easy-to-read description of the questionnaire with syntax removed and replaced by readable English. Now it gets created (<study>.hrd) each time you compile whether you have a "!hardcopy" command or not. You can use the {!-hardcopy} command to turn off the hardcopy output for certain sections of the questionnaire or use {!hardcopy <options>} to change the looks of the hardcopy file.

You can also use "~prepare compile -hrd" to negate the creation of a .hrd file., if you choose

DATA CLEANING, VARIABLE DEFINITION, TABLE-BUILDING SPECS (CLN, DEF, TAB)

If you also have the Mentor software, you can create cleaning and table specs during the compile. The CLN file, in combination with the CLEAN.SPX model spec file, will let you clean the data by the questionnaire's logic and rules. The DEF and TAB files, in combination with the TABS.SPX model spec file, will let you run tables automatically without you having to write the specs. (See your *MENTOR* manual for more details. See *2.3.1 COMPOSING AND COMPILING QUESTIONS IN PREPARE* for how to create these files.)

The DEF file is also where other program table definitions are stored if specified on the COMPILE line. This includes SPSS, SAS, and SSS_XML specifications.

A NOTE ON RENAMING FILES

PREPARE files are not automatically overwritten. If one exists when a new one with the same name is created, the old one is renamed by changing the first letter to the next ASCII character (DEMO.DB, for example, would be renamed EEMO.DB).

To replace rather than rename files, the PURGE_SAME meta command must be invoked. *Be careful* not to lose needed files using this command. The QUO file and the TR file are not renamed or purged by recompiling. The List File is not affected by a PURGE_SAME command –– use a dash before the name to get the same effect, eg. "-listfile".

2.3 CREATING PREPARE SPECIFICATIONS

PREPARE specifications include all the text, formatting commands and logic conditions that are necessary to run an interview with Survent. The question text is typed as it will appear on the screen during an interview. All other commands must follow specific syntax.

This section explains the syntax for writing PREPARE specifications. You can use this syntax to create an ASCII file and then call the file into PREPARE for processing or specify the commands line by line in the PREPARE program.

The basic syntax for questionnaire specifications is:

```
>PURGE_SAME
~PREPARE COMPILE
[Study header]
{question 1}
{question 2}
.
.
>Meta commands (optional)
.
.
{!Compiler commands} (optional)
.
.
```

```
.
{question n}
~END
```

>PURGE_SAME tells the program to replace rather than rename same-named files that are created in this run.

~PREPARE COMPILE tells the program to compile the specifications.

The Study Header describes the study name, case length and other parameters that control the overall interview process and how data is collected.

Question specifications display text to the interviewer and describe input data characteristics or serve as controls for the questionnaire flow or data collection.

Meta commands control the programming environment.

Compiler commands control general features of groups of questions or the questionnaire as a whole.

~END ends specifications and causes them to be compiled. Any commands after the ~END will be ignored.

In addition to the basic syntax, special characters such as the ampersand, used for continuation of multiple-line commands, and the apostrophes used for comments, are used. Their use will be discussed later.

2.3.1 Composing and Compiling Questions in PREPARE

PREPARE is a subset of the CfMC Mentor program. To use PREPARE, you must specify that you are in the questionnaire preparation mode. The command to do that is ~PREPARE. This must be invoked before any other questionnaire creating commands.

Commands preceded by a tilde (~) are CfMC commands specifying which programming level you are working in. In the PREPARE subsystem, ~PREPARE is the only level.

The ~PREPARE command may be abbreviated as ~PREP. You will find this type of abbreviation of commands available throughout the program. In this manual, we will use the long form of the command; abbreviations are shown in *Appendix B.*

There are two ways to use PREPARE to create your questionnaire:

1 Compose questions using menus and icons in the Script Composer.

2 Write specifications in a file using your own editor, compile the specifications with PREPARE, check for syntax or logic errors, edit the specifications, re-compile, etc.

Each of these has advantages. When using Script Composer mode, you don't have to specifically learn any question syntax. Syntax errors are almost impossible to make. Options are presented to you at each step. The question list is an easy reference tool and compiling and modifying specifications is simplified. The disadvantages of the Script Composer are: it is sometimes harder to specify multiple question specifications, and, at times, you have to flip from menu to menu.

Using an editor means you don't have to read the menus, you can use your standard editor or word processor (with which you may be more comfortable) and using programming techniques, you can copy questions, repeat a set of functions, etc. in ways not available in the Script Composer. You may get a better understanding of how the program works by learning the specific syntax.

In either case, you should read the manual for the contents of each part of the program. If you decide to use Script Composer in all cases, you need not concern yourself with the specific syntax. We are assuming you are writing question specifications for purposes of this manual, though.

To use an editor to create PREPARE specifications:

~PREPARE COMPILE should be the first line in the file (after the optional, but very useful, >PURGESAME). It directs the program to compile the questionnaire after the specifications have been entered and the program sees a ~END statement.

You can control the backup specification files that are created when you compile the specifications. By default, or by specifying ~PREPARE COMPILE SURVENT_SPECS, a backup QSP file is created with commands in standard Survent syntax. ~PREPARE COMPILE -SPEC_FILES compiles the questionnaire but does not create any files other than a QFF and QUO. This reduces the compile time somewhat.

Other keywords supported are QUANTUM, SAS, SPSS and SSS_XML to create files for other systems.

(See *Appendix F: CONVERSION TO OTHER LANGUAGES* for sample output using some of these keywords.)

By default, PREPARE creates the QSP, CHK, SUM QUO, QFF, and DB files. The HARDCOPY (.HRD) file is controlled by the !HARDCOPY keyword in the questionnaire. By using compile keywords, you can get other program-generated specs created:

```
SURVENT_SPECS: default QSP, CHK, SUM, QUO, QFF, DB files
-SPEC_FILES: suppresses all files except QUO and QFF
MENTOR_SPECS: DEF TAB
QUANTUM_SPECS: DEF TAB
CLEANING_SPECS: CLN
COSI_SPECS: DEF
SPSS_SPECS: DEF
SSS_XML: DEF
```

NOTE: Keywords that write to the same spec files cannot be used together. The QUO, QSP, CHK, HRD, DB and SUM files can be suppressed by options on the header.

The study header is next, followed by question specifications.

~END causes the specifications to be compiled and exits you to the operating system prompt.

A PREPARE compile consists of up to three steps:

- Syntax checking
- Label resolution
- Variable conversion

Syntax checking is done on each question and command as the file is read. Error and warning messages are displayed at the screen and printed in the list file. You will see an indication of each

question as it is read; labeled questions will display the question label, and questions without a label will print their assigned label. Error messages are printed like “ERROR #### text...”, and warnings “WARN #### text...”.

Label resolution guarantees that all references to other questions are to actual questions of the proper type for the reference.

Variable conversion makes DB variables for later use by the CfMC utilities or Mentor program (if a DB file is being created). The auxiliary files (DEF, TAB, etc.) are also made during this phase.

CONTROLLING THE AUXILIARY SPEC FILES CREATED

You can compile Survent specifications with the -SPEC_FILES option (makes only the QFF and QUO files) and produce only those auxiliary files you need once you are satisfied with the questionnaire.

```
EX: ~PREPARE COMPILE -SPEC_FILES
```

Any auxiliary file can be generated from a compiled QFF (questionnaire) file without re-compiling. The syntax for generating auxiliary spec files from a compiled Survent questionnaire is:

```
~QFF_FILE <studyname>
~PREPARE MAKE_SPEC_FILES keyword
```

where the keywords include all those available under ~PREPARE COMPILE (see previous page) and these additional keywords:

FILES CREATED

CHK	CHK
HRD	HRD (if {!HARDCOPY} is specified
QSP	QSP‘
SUM	SUM

```
EX:
~QFF BANK    (Loads the compiled QFF file.)
~PREPARE MAKE_SPEC_FILES (Generates a CHK file.)
```

NOTE: If there is a password in the study header, you must know it to generate files.

When using PREPARE, auxiliary/non-required files are generated after the compile. Consequently, for the fastest compile, use ~PREPARE COMPILE -SPEC_FILES.

2.3.2 The Study Header Statement

This is always the first statement following the ~PREPARE COMPILE command. It provides PREPARE with the study name and general information related to the questionnaire.

The study header statement is enclosed in brackets. It consists of a study name and any desired options. The only required item inside the brackets is the study name. Commas or spaces must be used between items.

The syntax for the study header is:

```
[study name,options]
```

The **study name** is used by PREPARE to name the various output files. Survent will also use the study name to name the data file in the interviewing process. The study name may be three to 30 alphanumeric characters long, starting with a letter. Keep the name shorter for cross-platform compatibility.

The **options** are used to control many questionnaire functions. There are interviewer-related, operational, supervisor, size-related, and compile-time options. You may include as many options as desired in any order, placing a comma or space(s) between each option. Ampersands must be used to continue the line if the header statement is longer than one line. The available options and their functions are:

INTERVIEWER DISPLAY AND CONTROL OPTIONS

ALLOW_ABORT

Allows the interviewer to abort an interview by typing ABORT. You can also use the compiler directive ALLOW_ABORT to turn this ability on and off throughout the interview. Specifying this option on the header merely changes the default setting.

ECHO_CATS=option

Places you in Echocats mode on code lists. Responses to all CAT and FLD questions will be echoed (response code highlighted) when the interviewer presses **Enter**; an extra **Enter** is required to move to the next question. (See *4.2.2 RESPONDING TO SPECIAL QUESTION TYPES* for more details.) ***NOTE:*** This will override the HIGHLIGHT_CATS compiler command. You can also control this by using the compiler command ECHO_CATS or by assigning the interviewer this capability in the employee information file (E in columns 42-44). Specifying this option on the header merely changes the default setting.

The option can be SINGLE, MULTI or ALL. This controls whether Echocats will be done on single-response questions only, Multipleresponse questions only or all questions (default).

ERROR_BEEP

Alerts interviewer with a beep when an invalid response is encountered; an error message also appears on the screen. The default is that the error message is displayed without the beep sound.

FLAG_DISALLOWED_CATS

Shows all responses in the response list for CAT and FLD question subtypes C and D and I, but highlights any that are not allowed as responses. (See *2.4.1 CATEGORY QUESTION TYPE*.) Also available as acompiler command to turn on/off during the interview.

HIGHLIGHTCATS_MATCH_TEXT

When using HIGHLIGHT_CATS mode on !CAT or !FLD questions to use an interactive screen on a code list, this says to match text the respondent enters to the TEXT of the responses rather than theresponse CODES. This is very useful if your code lists are alphabetically sorted. You can use subtype "M" on a particular !CAT or !FLD question to invoke this for that question only. This header option would apply to all code questions using HIGHLIGHT_CATS mode in the questionnaire.

INFO_BETWEEN=#

This specifies what to print on the screen between interviews. The options are additive, that is, "5" would do options 2 + 3, "6" would do 1,2 + 3, "17" would do 16 + 1, etc. The default is "11", show the # completes, time and interviewer info. The base options are:

0	shows nothing
1	shows the number of starts and completes for that session
2	shows the current time and time since last interview
4	shows the time since start of session
8	shows the interviewer ID device number
16	shows phone info (# records, # calls made, # completes

KEEP_BEFORE_PHONE_9

This header keyword lets you control what to do with questions prior to the !PHONE,9 statement when resuming. The default is to keep the old answers (the ones answered previously). If the option is turned “on,” then you’re asking the program to save the responses given in the current interview instead of the original. Be careful when using this because it could affect questionnaire logic if the questions in the main part of the questionnaire are based on actions done by questions prior to the !PHONE,9.

LANGUAGE=

The LANGUAGE= header statement is where you specify which languages will be included in this questionnaire. It also allows you to specify the character set and the starting language. This MUST be included if you have multiple languages with the \Lxx language control specifications on your questions (see \L for more information).

The languages being used can have two-character codes or can have the two-character codes converted to a one-character code for ease of coding, for instance:

```
     EX:
LANGUAGE=(set=(e=en f=fr s=sp) Character_set=Extended_Ascii Speaking=E)
```

This says to use English, French, and Spanish, and allow coding of them as one-character codes in the questionnaire (eg. \le, \lf, \ls).

The "character sets" are ASCII, Extended_ASCII, Shift_jis, or Multi_byte.

The "Speaking=<language>" parameter would be used if you want to change the current language; if you don’t set it, it defaults to the first language on the list.

LANGUAGE= now supports universal character set UTF8, allowing all languages in the same questionnaire. The questionnaire header option and compiler directive LANGUAGE= supports this character set that can be used for ANY language and is used by "UNICODE" language coding. If you use this, you do not have to distinguish between MULTIBYTE, EXTENDED ASCII, and SHIFTJIS character sets, and they may all be used in the same questionnaire.

This, along with the ability to increase question size, means that you can have one questionnaire with all your languages included instead of having separate questionnaires for each.

RESUME_LOOKAHEAD=#

This controls how far ahead to look for changed or unlabeled questions when resuming. It controls how far ahead to look to find questions without labels that have moved as a result of questionnaire changes you made when trying to resume an old version of the questionnaire. The default is that it will look ahead up to 50 questions before "fail"ing.

ROTATE_SEED_WIDTH=#

This controls whether the !ROTATE,S statement uses a "seed" that is 6 wide or 10 wide. The default is now 10 wide because it was found that all combinations in big rotates were not allowed; we are allowing users to set this to 6 for back compatibility to old programs. Use this if you need to retain the same width as earlier versions when recompiling with version 7.7.

There is also a new command ~set rotate_seed_width=6/10 which will allow you to control this across all studies compiled. Only use this to compile old versions of questionnaires where you need to retain the shorter width.

SHOW_LAST_RESPONSE

This shows the response of the previous question at the bottom of the screen of each question. This is useful so the interviewer can remember the response to the prior question.

SHOW_QUESTION_LABELS

Prints question labels in the lower right corner of the screen during an interview. The default is to not print them unless you are DEBUG or PRACTICE mode. For doing this in standalone interviewing under DOS refer to section *4.1.1 MODIFYING THE CONFIGURATION FILE* item 6.

Also see the same named compiler command in *3.2.1 INTERVIEW CONTROL COMMANDS*, that will let you toggle the display of labels during the questionnaire.

Specifying SHOW_QUESTION_LABELS=UR will let you place the question label in the upper right corner of the screen instead of the lower right corner.

SOUNDFILE_FORMAT=

The keyword "SOUNDFILE_FORMAT" controls whether you get a fully qualified name or other formats. This can be specified in the CfMC parmfile or in the questionnaire header. The possible values are:

- EIGHT_DOT_THREE: This will generate a name that the gerry dialer can accept which has Sjjj (julian date), then hhmmss compressed to fourcharacters using base 36, plus a three-character compressed ldev extension.
- JJJHHMM: This will make a soundfile name Sjjjhhmm. where jjj is the Julian day and hhmm is hour and minute. This is the old default, but it has a problem: if you have twofiles that start in the same minute on the same question, it will overwrite the first with the second, and you can't have an LDEV with a number higher than 999.
- YYMMDDHHMMSS: This is the default using Winsound and will make a name "S._" which has the year-month-day-hour-minute-second laid out clearly. It will also include the interviewer ID (four characters) and ldev (five characters) as part of the extension.

If there is no parmfile keyword, the soundfile format will be "EIGHT_DOT_THREE" if you are using the Gerry dialer for sound recording and "YYMMDDHHMMSS" if you are using Sound Survent.

TEXT_HELP

Displays "F1 for help, ESC to exit" in the bottom line of a TEX question response box (DOS, UNIX only) so the interviewer knows what to do at that question type.

TIME_ON_SCREEN=<LINE#>

Shows the time and time since the start of the interviewing session in minutes, right- justified on the line specified. This is often useful information for interviewers.

```
EX:
                                                  10:23 15
```

Would display on the right side of the screen if this is specified, saying the time is 10:23 a.m. and the interview has been running for 15 minutes.

SURVENT OPERATIONAL OPTIONS

-AUTOMATIC_CASEID_INCREMENT

Says to not have Survent automatically provide case IDs for the data file. Interviewers will be prompted to assign an ID.

CARD_ID_FORMAT

Puts the case ID on every 80 column record of the file (in the columns specified on CASE_ID below), and a record ID in the location specified here. The record ID generated is a number starting with 1. Specify a width greater than one if you have ten or more records. If a width greater than one is specified, the first column(s) of the record ID will be zero-filled (e.g., 01, 02). You can also add + *col.wid="comment"* to put a comment (maximum 10 characters) on every 80 column record in the location specified. CARD_ID_FORMAT items will not be written in the TEXT_START area or the WORK_START area. The default is to have no record IDs, and have the case ID in only one place.
CASE_ID=col.wid

Places the case ID in the specified column location, with the specified width. The default is to put it in column 1 for a width of 4. The allowable range for the width is 3 to 10. Unless CARD_ID_FORMAT is set, the case ID will appear only once.

COMMENT="text"

Allows up to a 49-character comment which will display when an interview is started. The text must be enclosed in quotes. The comment can also be specified as a quoted string directly after the study name and case length without a keyword.

FAILED_RESUME=option

Determines how to handle changed questionnaires when they are resumed and the changes cause the new and old version not to match. This overrides the PARMFILE option of the same name and tells the program what to do when it tries to resume the interview.

The option can be BLOW, STARTHERE, or RESTART. BLOW causes the program to blow when a suspended interview does not match the current interview; this saves the data collected so far in a separate data file to be reviewed later and aborts the interview. STARTHERE will restart the interview from the point that it found a discrepancy.

RESTART will do the default, that is, re-start the interview from the beginning.

INTERVIEWER_LOGGING

Interviewer_Logging or Interviewer_Logging=yes will cause every interviewer on that study to be logged. See *4.6 SURVENT LOGGING* for more information on logging. Normal logging will save the log file after the completion of an interview.

You can also use the shorthand “Log”. Specifying Log=After_Every_Question will save the log file immediately after every question, so that if you are trying to find a problem (where the program is aborting abnormally for some reason), you can save responses up to that point and try to reproduce the problem. Note that this may slow down your system if you are on a multi-user computer and many people are using this feature.

Specify Log=No or –Log to turn logging off for this study, even if logging is turned on for your site in the CfMC PARMFILE. You can control this from the supervisor by using the “LOG” and “STOPLOG” commands. You can control this from the interviewer station by using the LOG at the "Return to interview" prompt. Logging after every question can be invoked at the "Return to interview" prompt by entering "LOGDEBUG". Logging may be turned off at the prompt by specifying “–LOG”.

NEXT_CASE_ID=#

Sets the initial case ID in the data and quota files, or sets the next ID if the data file already exists. # must be an integer number and be greater than the current value, and must be the requisite number of digits.

ONE_INTERVIEW

Causes Survent to start the interview without the standard interviewer prompt and to exit after one interview is completed. Useful for sending out diskette mailings that require one interview as a response.

QUOTA_CRC_CHECK

Controls whether to check the quota file for a valid names block. The default is to check that the quota file's quota name block matches the block that was compiled with the questionnaire. Say -QUOTA_CRC_CHECK to override this and not check. Use this if you have one questionnaire that had quota updates and others running at the same time with the same study name that didn't. If you use this to turn off CRC checking, you run the risk of updating the incorrect quota if you make changes to the quotas in the questionnaire without updating the quota file as well.

REDO_SPECIAL_BLOCK_ON_RESUME

This is the default, it controls whether to re-execute the “SPECIAL” block when resuming the interview. If you are using the “SPECIAL” block to do actual work like edit questions, you will want to do this, otherwise you may not if for instance, you are using it to call a specific telephone number or as a help block.

TEST_ONLY

This creates questionnaires that can be tested by Survent and used for REFORMAT or other programs, but it will not generate data. This is particularly useful so that CfMC can provide all clients with the ability to compile questionnaires in DOS to send to their clients for evaluation, without requiring a security dongle be connected to the computer. It can also be used to put questionnaires up for testing and guarantee that users will not generate test data by mistake.

TIME_QUESTIONS

This will record the time between questions in the server log file. The format of the log record saved is:

```
    Column      4         5         6
LL record format: 01234567890123456789012 3
time on qq: qstlabel ###
```

```
01-04 LL16
05-39 Standard LL record info (time/interviewer/study)
40-50: "time on qq:"
52-59: question label
61 : time on question in seconds
```

You could then run reports on these values to get the average time between questions.

TIMESTAMP

Requires that the QUO file built or updated during this compile be used for interviewing. See *3.1.5 QUOTA INCREMENTING STATEMENTS, Updating the Quota File* for more information. If this parameter is used and the wrong QUO file is used, Survent will not run.

USE_DUMMY_IVR

Required if the questionnaire will be using the “Dummy interviewer” feature to post-process phone numbers returned from a predictive dialer.

SUPERVISOR/MONITOR RELATED OPTIONS

ALLOW_VIEW

Allows View mode without requiring knowledge of the study password, so users can view and alter data without allowing users to change phone parameters or quotas (see MODFONE and MODQUOTA commands).

COUNT_AS_COMPLETE=#

This allows you to specify up to 5 statuses to be counted as completes in the SUPERVISOR DAI menus. Use “COUNT_AS_COMPLETE=W” to count every case written with a case id as a complete. A typical use would be to specify "COUNT_AS_COMPLETE=1" to count status "1" as a complete.

Note that the server and phone file counters count completes as phone status "1" no matter what, but the log reports can be adjusted to count completes however you'd like.

DASH_BOARD

Specifies to update counts on the DAI STATS screen in the supervisor for this study. Note that this requires additional overhead for the CfMC Server.

MODIFY_CASE_ID=Y/N/P

Specifies the supervisor's (SURVSUPR) access to modifying the "next" case ID. If Y(es), SURVSUPR can make changes using the MODID command. If N(o), SURVSUPR can't, but the CLEANIT utility can using the NEXT_CASE_ID command. If P(ass) and the PASSWORD header option is specified, the supervisor will be able to make a change to the ID if they know the password (they will be prompted). The default is Yes.

MODIFY_FONE_FILE=Y/N/P

Specifies the supervisor's (SURVSUPR) access to modifying the FON file. If Y(es), SURVSUPR can make changes using the MPF command. If N(o), SURVSUPR can't, but the FONEUTIL utility can using the M command. If P(ass) and the PASSWORD header option is specified, the supervisor will be able to make changes to the FON file if they know the password (they will be prompted). The default is Yes.

MODIFY_QUOTA_FILE=Y/N/P

Specifies the supervisor's (SURVSUPR) access to modifying the QUO file. If Y(es), SURVSUPR can make changes using the QSS MOD command. If N(o), SURVSUPR can't, but the QUOTAMOD utility can using the S MODIFY command. If P(ass) and the PASSWORD header option is specified, the supervisor will be able to make changes to the QUO file if they know the password (they will be prompted). The default is Yes.

-MONITOR

Controls monitoring from SURVMON. If you don't say -MONITOR, you can monitor (default). If you say -MONITOR and PASSWORD=, you need to specify the password to monitor. If you say -MONITOR and don't specify a password, you cannot monitor. This has no effect on SURVSUPR's MON command.

PASSWORD=

Allows you to assign a password to access View mode in Survent, the MPF, QSS MOD, and MODID commands in SURVSUPR, and monitoring in SURVMON. There is NO password required by default. The password can be from 1 to 8 alphanumeric characters and must be enclosed in quotes. See also the SUPER_PASSWORDS PARMFILE command, also see note at end of section for more information on passwords.

TRIPLE_QUOTAS

Creates three quotas for each quota name referenced in the specifications: a standard quota (name), a target quota (name.T), and a resettable quota (name.R). Also allows the use of the TARGET compiler command to set the initial value for the target quota.

You can use "TRIPLE_QUOTAS=MOD_TARGETS_ONLY" and then supervisors can only modify the target quotas on QSS screens; they won't have access to modify the other quotas.

PROGRAMMING/SIZE-RELATED OPTIONS

ANSWER_LENGTH=#

Sets the size of the "answer array" to # bytes. The default is 65000. The maximm is 500000. This is the approximate number of characters allowable in back-references to questions and is only needed if you have many back-references.

CASE_LENGTH=#

Specifies the number of columns in the data file for the respondent's answers. Can also be specified immediately after the study name without a keyword. This may be specified using an absolute column number or using the syntax record/column, where each record consists of 80 columns (e.g., 800 or 10/80). The default is 800 columns, the maximum is 32000 and the minimum is 10.

FONE_TEXT_LENGTH=#

Specifies the number of columns of text in the phone file. The default is 200, the maximum is 4900). PREPARE will check that this boundary is not exceeded in PHONE,G and PHONE,P statements and \[phone references. If set to a number larger than that specified in your phone file, Survent will blow.

MAXIMUM_GRID_QUESTIONS=#

This allows you to have more than the default of 200 questions in a grid. There is no maximum, but if you use this,, you may have to increase the amount of core memory survent runs with. To increase the core memory, say “survent server core:2000000” for instance.

MAXIMUM_JOURNAL_FILE_SIZE=#J

This keyword can be used in cases where the “response” file for a survey exceeds the default limits. By default, the journal file is 15000000 bytes. The absolute maximum you can set it to is 536870000 bytes. If you run out of space in the journal file and you receive a warning message, you can update the maximum size by using this keyword.

MAXIMUM_LABELS=#

Allows you to set the # of labels allowed to a smaller value to impact the size of the compiled questionnaire file (QFF). The default 64000 labels, setting it smaller makes a smaller questionnaire file. 1,000,000 is the maximum allowed. This means that you can do almost use any application without having to split the questionnaire up into multiple !CALL questionnaires.

MAXIMUM_QFF_FILE_ SIZE=#

Specifies the maximum size in megabytes for the QFF file. The default is 1 megabyte and the maximum value is 50 megabytes. **NOTE:** Your hardware must also be able to work with the QFF file produced.

MAXIMUM_QUESTION_SIZE=#

Allows you to have larger questions, particularly for use with long brand lists, multiple languages or webSurvent. The default size is 64000 bytes per question. This has the following applications:

10,000 CODE LISTS: This will allow up to 10,000 codes and 750 responses on one question. In particular, this is most useful for a controlling question for a list of cars, etc. where you use !FLD,I to include or exclude codes from that question.

10,000 LINE DISPLAYS: You can have up to 10,000 lines of display text (take note that browsers may be limited in the amount they can display).

MAX_QUOTA_NOW=#

MAX_PHONE_M=#

These options enforce rules for the number of file updates per interview. There is now a maximum number of updates of the quota or phone file that you can do in a single interview. By default, the maximum !QUOTA,2 or !QUOTA,1,name,NOW updates you can do is five and the maximum !PHONE,M updates (to change a market weight) you can do is one per interview.

If you are doing more than this, you are probably doing it by mistake, and it would slow the CfMC server down considerably. You will get a "BLOW" error if you try to do more than the maximum number of updates and you will have to modify the questionnaire if you want more.

To allow more of these, you can set a higher maximum in the questionnaire header using the "MAX_QUOTA_NOW=#" or "MAX_PHONE_M=#" commands, but beware that this can put a heavy load on your CfMC server.

MAXIMUM_QUOTA_NUMBER=#

Specifies the highest number allowed for numbered quotas. This must be used if numbered quotas are used. # can be a number as small as 100 or as large as 2,156,000,000. The QUO file will grow by 4 bytes for each number specified here.

NOTE: If you specify the maximum number, you will be building a quota file of approximately 40 megabytes in size.

ROTATE_SEED_WIDTH=#

Specifies the allowable size of !Rotate seeds for !Rotate,S statements.

The default is 10, but this lets you set it to the old value of 6 to allow older spec files to retain the old value of six characters. The maximum was increased to 10 to accommodate more questions per rotate. This ensures that all possible combinations of larger rotates would be random. With a six-digit random seed, this could not be guaranteed.

SCREEN_LINES=#

Specifies the maximum screen size. The default is the size of the standard screen on the machine you're compiling on, which is 23 for UNIX and 24 for DOS. If a question has more text lines than #, the interview will be aborted with an error message. Note that this does not include the text from the response list; if the total screen size including the response list is greater than #, it scrolls the screen until a prompt for the response can be displayed. WebSurvent users should set this to 200 or so.

SPEC_WIDTH=#

Defines the maximum specification line width and controls the width of the QSP, SUM, DEF, and HRD files. The default is 5000, and this is the maximum as well. The minimum is 72. Set this to 80 if you want to check that all screens will fit on an 80 column screen.

TEXT_START=#

Stores the data from TEX questions beginning at the location #. The text area is set at the end of the WORK (see WORK_START) or regular data area and goes to the end of the data record. This must be at least 100 columns less than the case length. The default starting location is the first column of the next blank 80 column record. The user cannot assign data columns above this point.

TRFILE_DIRECTORY=#

This controls the size of the datafile case id index when the datafile is created by Survent. The default is 20,000 entries. The index is used by coding applications to find the record wanted quickly, or in the cleanit utility to find cases quickly. If you expect more than 10,000 cases you should increase this value. To remove the caseid index for this study, say –TRFILE_DIRECTORY. Note: You can set this for your site in the CfMC parmfile.

WORK_START=#

Allows you to specify the starting location of a data area where the system will not change the next data column pointer after questions where locations are specified. It will also not check for data column overflows or warn you about the next data location being less than this.

You must specify a data column for each question in the work area, or use the keyword “WORK” on a question to put it in the next available work area location.

This is usually used as a place to put data generated from the interview or to store TEXT question pointers or ROTATE seeds if you don’t want them mixed with the rest of the data. TEXT_START=# must be greater than WORK_START=# if using that option when you have TEXT questions. The ending location of the work area depends on the TEXT_START or CASE_LENGTH.

PROGRAMMING/COMPILE RELATED OPTIONS

ALLOW_DATA_OVERLAP

This command allows you to have questions that start in the data area of a prior question and go beyond the end of the data area for that question, or that start before a question and overlap into another data area. The default is that this is not allowed, because you are probably mistakenly overwriting data.

-CHECK_FILE

Suppresses the creation of a list of the compiled questions and their data locations (the CHK file).

CHECK_FILE_INCLUDE=

Allows more items to be included in the in .chk file. It allows you to include two question types that were previously excluded from the .CHK file (because they don't GENERATE any data). The first is the subtype "A" questions (CAT, FLD, VAR, TEXT). These are used to display responses and can now be reported in the .CHK file. The second includes are "hidden" questions. These are generally used to control logic or they can be used as placeholders for table definitions. You can aso include these in the .CHK file output. The syntax is:

CHECK_FILE_INCLUDE=ATYPE/ALL/HIDE

You can use shorthand like "CHKI=ALL" to include all types.

DATA_LOCATION_REQUIRED

Requires all data-asking questions to have a data location specified (or a previous question's location). This is useful when hard coding data locations, to ensure you don't mistakenly let the program assign any data locations for you.

DATA_OUTSIDE_LOOP_OK

This keyword is necessary if you want to have some LOOP variables save data outside the LOOP area, since this is dangerous and could cause data overwrites. The default is that data is not allowed outside the LOOP area.

-DB_FILE

Suppresses the creation of a variables data base (DB) file.

-HARD_COPY_FILE

Suppresses the creation of a hardcopy (HRD) file. This will override a HARD_COPY compiler command in your spec file.

MAKE_NAMES

Tells the program whether to make names for variables not otherwise named by the programmer. This is the default. Use –MAKE_NAMES to take it off. Variables are named "cfm#####" where the number is incremented by one for each new name.

NUMERIC_WIDTH_REQUIRED

Requires a width on all data location specifications for all EXP questions and NUM questions without a range. This keeps you from accepting the default width of 9 columns for these.

QFF_FILE_NAME=name

Specifies a QFF file different than studyname.QFF. This causes the questionnaire file produced by PREPARE to have the name designated here instead of the default name. One use of this is to set up a coding questionnaire with a unique QFF file name, but using the same data file being interviewed on by someone else. This QFF file will still use the FON, QUO, and TR files as the study name implies.

-QUOTA_FILE

Suppresses the creation of a quota file (QUO). To have shared files, Survent must have a quota file. It will also keep the CRC check from occurring (see QUOTA_CRC_CHECK). Use this if using !CALL questionnaires with no quota updates in them (the program will use the MAIN questionnaire's quota file).

QSP_FILE

Suppresses the creation of the backup specification file (QSP) at the time of compile and possible later decompile (~PREPARE MAKE SPECFILES).

SPL_DATA_LOCATIONS_OK

Allows the use of HP3000-SPL specification type format for data locations e.g., A01 for record one, column one, or C23 for record three, column twenty-three. Otherwise, you need to use absolute column (183) or record/column (3/23) format. Also allows A01 type references without brackets around the location specification on IF conditions (e.g., A01^1 rather than [1/1^1]). If you use this option, question labels cannot be in this format (i.e., Q23 is a bad label).

-SUMMARY_FILE

Suppresses the creation of a summary list which contains a one-line description of each question (the SUM file).

VALID_WIDTH_REQUIRED

This forces the questionnaire writer to match the proper width the program would assign on all questions. Otherwise, users can make widths that are greater than or equal to the required width.

To see options that affect ALL studies, see the CfMC PARMFILE options in Chapter 4.

Here are some sample study headers:

```
      EX: [TEST1]
```

This is the shortest valid header statement, since only a study name is required.

```
EX: [TEST1,CASE_LENGTH=5000,ANSWER_LENGTH=25000,ERROR_BEEP]
```

This study header includes three options (which can be entered in any order).

```
      EX:
 [TEST1,CASE_LENGTH=1000,COMMENT="My First Questionnaire”, &
ALLOW_ABORT]
```

TEST1 is the study name, which will be used when PREPARE and SURVENT create files or refer to the study. The data file length will be 1000 columns. *My First Questionnaire* is the study comment.

ALLOW_ABORT will allow interviewers to abort an interview.

```
      EX:
      [TEST1,CARD_ID_FORMAT=5.2+7.6="123-45"]
```

CARD_ID_FORMAT will put the case ID (in columns 1-4, by default) and record ID (in columns 5-6) on every record as well as the comment “123-45” (in columns 7-12).

```
EX:
[TEST1,CASE_LENGTH=5600,COMMENT=Questionnaire", &
ALLOW_ABORT,TEXT_START=1501]
```

Note the ampersand at the end of the first line of the example. TEXT_START= 1501] is a continuation of the first line and ends the study header.

2.3.3 Question Statements

After specifying the session controls (~PREPARE) and questionnaire controls (header statement), you need to tell PREPARE about the specifics of the questionnaire that you wish o use for interviewing. PREPARE uses question statements, compiler commands, and meta commands to control this.

Question statements are used to collect the data or perform other functions. They are discussed below. Compiler commands are used to turn on or off compile options, question control options, and data control options. They are discussed in *3.2 QUESTIONNAIRE CONTROL USING COMPILER COMMANDS.* Meta commands affect the programming environment. They are discussed in *3.3 PROGRAMMING ENVIRONMENT*.

You can have as many question statements, compiler commands, or meta commands as you like in a questionnaire, limited only by data space (6240 columns for 286 DOS or 25,000 for others) and the time it takes to complete an interview. To overcome the data limitations, you can use the CALL statement to call other questionnaires.

There are two kinds of question statements used by PREPARE:

- **Data entry** questions are statements that display a screen and prompt the interviewer for a response. The specific question type tells PREPARE what kind of data should be allowed as a response, whether to present a list of choices, how to edit the response, and how to store the data. See *2.4 DATA ENTRY QUESTION TYPES* for information on the different question types.

- **Control** statements are constructed in the same manner as data entry questions, but they are not used for data collection. Instead, these statements are used to control the interview process in various ways, display information to the interviewer, or do internal data creation. See *3.1 CONTROL STATEMENTS* for more information on these statements.

Question statements use the specific syntax explained below. The action taken is controlled by the question types. Questions may also be individually controlled according to their data locations, whether they are to be asked based on previous questions, how they are presented to the interviewer and the type of data that is recorded.

2.3.4 Question Structure

Each question specification is enclosed in braces ({ }). The left brace and question label line options are on the first line, followed by lines that specify other parts of the question structure. A right brace follows whatever is on the last line of the question specification, either on that line or on a line by itself.

The syntax for a question is:

```
{label: location options (required)
!HELP, MISC, or RDG statements (optional)
!IF condition statement (optional)
question text lines (optional)
!question type, subtype, options (required)
response list items (if required)
} (required)
```

The first line is the question label line, which includes a brace ({), a label, a data location and other options. Only the brace is required. A label or number is required if you want to reference this question from other questions. You can use a minus sign before the label,(eg. {-label: ...} to comment out the question (see *2.3.5 QUESTION LABEL LINE*). The !HELP statement allows you to specify a line of text that will print at the bottom of the screen for this question; this will be discussed later. The !MISC statement allows you to set certain values for purpose of generation of supporting files (also discussed later). The !RDG statement controls the frequency of certain responses when doing “Random Data Generation” questionnaire testing (See *2.7.2 RANDOM DATA GENERATION*).

The condition statement is on the next line. This is included if you need to specify that the question be asked only under special conditions (see *2.6 CONDITION STATEMENT*). An individual condition statement can have as many as 32,000 characters of references. This allows a virtually limitless condition so that you don't have to worry about breaking it up into separate questions.

The question text begins on the following line. The text of the question is free-form; it will be presented to the interviewer exactly as you format it. Special screen formatting commands are available to highlight words, display prior responses, etc. (see *2.5 QUESTION TEXT*).

The question type, subtype and options are specified on the line following the question text. The available subtypes and options depend on the question type and are described in the sections on question types and control statements (see *2.4 DATA ENTRY QUESTION TYPES* and *3.1 CONTROL STATEMENTS*). A question type must be specified.

For some question types, a response list must follow the question type, listing the possible responses. You can control the data stored and skip to other questions based on the response. Thisis described in sections *2.4.1 CATEGORY QUESTION TYPE, 2.4.2 CATEGORY QUESTION RESPONSE LIST*, and *2.4.4 FIELD QUESTION RESPONSE LIST*.

Only the question label line opening brace, question type, and closing brace are required. Depending on the question type, other parameters may be required. We will discuss the required elements of the questions first (question label line and question type) followed by the optional elements (question text and condition statements).

For advanced users:

Where appropriate, you can specify a question structure on fewer lines. One-line items such as GENs, GOTOs, or QUOtas can be moved up to the label line.

```
EX:
{!GEN,M,to,from,length}
{!GOTO,label}
```

Here is what you are allowed to move up, and when:

- If your question label line is blank, or has only a label: on it, you can move up the next line if it is a ! line (not text).
- You can never move text up.
- The only two different items you can have on the same line are the question label and a !item.

This means you can also put your !IF on the question label line.

```
EX:
{INCMAN: !IF SEX(1)
```

If using the !MISC or !RDG or !HELP commands in your question structure, they are specified at the top of the question block, either before or after the (optional) IF, in any order.

2.3.5 Question Label Line

The question label line contains the label, question number, data location and width, and options, in that order. Each is separated by one or more spaces.

The syntax for a question label line is:

```
{label: [location] options
```

Note: For the syntax explanations in this manual, wherever label references are used, question number references are also allowed.

Question Label

The label is a name for the question. It is always followed by a colon (:). The label can be used by other questions to reference this question and for other functions such as skips and rotations. The question label can contain up to 30 alphanumeric characters. It may also include periods (.) and underscores (_). The first character must be a letter; the name cannot start with QQ. The label cannot end with a period followed by a number (i.e., ABC.1). This is because a label can be used as a data location specifier and that may require a width. In addition, it cannot end with a period or contain more than one consecutive period. Any label may not match the study name. A label is not required. However, CfMC recommends that you provide a label (or a question number) if the question will be referenced by another question. The label appears in the CHK and SUM files created after compiles and may print at the bottom of interviewers' screens.

```
EX:
{RESP_AGE:
```

PREPARE assigns temporary labels if you do not include a label. They are in the form of QQ###.##. You should not reference temporary labels from other questions because they will change as that question is accessed (modified, moved, etc.) or other questions are added or deleted.

PREPARE saves the question labels in the compiled questionnaire file for interviewers to reference. There is a limit of 4000 labels in a questionnaire.

If the questionnaire has a question with the special label of TERMINAT and the interviewer types "TERMINATE", the interview will skip to that labeled question and do whatever is indicated. This may include asking one or more questions, incrementing quota counters, saving the incomplete data case (or not) or terminating the interview.

```
EX:
{TERMINAT:
```

The Terminate block is typically placed at the end of the questionnaire with logic branching nonterminators around it. If you prefer multiple Terminate blocks in your questionnaire (i.e. doing different things at different points in the questionnaire), label them TERMIN_A, TERMIN_B, etc. When the interviewer enters TERMINATE, they will go to the next such block.

You cannot use question labels in the letter/digit format (i.e., Q10) if using SPL_DATA_LOCATIONS_OK on your header, since PREPARE may confuse the label with a data location and generate erroneous warning and error messages.

The following Survent keywords are disallowed as question labels: ABORT, ABORTOS, ALTER, BKOK, GOTO, RESET, RETAKE, SHOW and SUSPEND. Also disallowed are function names, such as X, NUMITEMS, QUOTA, etc.

You may use a minus sign before the label to “comment out” the question so it does not get executed (eg. {-label: …}.

Question Data Location

The data location is where the data is stored for data entry or data creating questions. The data location is not needed for other question types. The default data location is the *next* data location that is the next available location beyond the furthest specified location so far. By default, the data width will also be automatically determined, based on your question specifications. To avoid overwriting data, CfMC recommends that you not specify a data location or width unless it is necessary. If you do specify a data location, you should specify the location throughout the questionnaire (see header option DATALOCATION_REQUIRED). You can look at the CHK file to see the locations assigned to all questions, whether assigned by the program or by you.

The syntax for the data location is:

```
[label.width] or [column.width] or [column-column] or WORK
```

The keyword “WORK” will assign the location of the question to the next available location in the WORK area (see header option WORK_START). Using this keyword, you can have “column-free” questionnaires, even if you want to separate your data into the "Regular" data (to be sent to the client, etc.) and the “Work” data (used for calculations, etc.).

```
EX:
{Q1Total: WORK
EXPR,,X(STORE1)+X(STORE2)+X(STORE3)+X(STORE4) }
```

Otherwise, the location specification may placed within optional brackets([ ]). If you use the column.width format, the brackets around the location are not required (although when referencing a data location on condition statements or control statements, it is always specified within brackets; see *2.6.2 USING DATA LOCATION REFERENCES*).

You may use a previous question's label as the data location, if, for instance, you want the data from two different optional questions stored in the same location. You may also specify a numeric location (90), or use record/column format (2/10). Records are divided into 80 columns each, so 2/10 and 90 refer to the same data column. You may specify a data location in HP3000-SPL format (i.e., A25), but will need to specify SPL_DATA_LOCATIONS_OK in the header statement. (Also see the meta command LOCATION_FORMAT.)

When using a label or question number for the data location, you can use a column offset.

Here is the syntax and an example for using a column offset:

```
[label+/- offset.width]

EX:
{[FIRST+3]
```

This example will use the data location three columns higher than the first column used by question FIRST. An offset of 0 means the first column in the field (same as using no offset).

If you specify a location without a width, PREPARE will provide the width (based on the question specifications). Labels without widths will use the same width as the original question.

```
EX:
{ [Counter]
```

This would put the data for this question in the same location as the question labeled Counter. The same width as Counter will also be used; you must specify the width if you want a different one.

You will be warned if the length given is less than or greater than the width needed.

```
EX:
{ 1/23
```

This would put the data in the location record 1, column 23. PREPARE will fill in the width.

```
EX:
{ 1/5-1/15
```

This would put the data in record 1, columns 5 through 15. Absolute data locations (5-15) could also have been used, as well as specifying 1/5.11.

The system's default printing format for data columns is record/column location. You can change this with the >LOCATION_FORMAT meta command. This would affect the format printed in the CHK file, the SUM file, the QSP backup file, and the HRD hardcopy file.

The width would be useful if, for instance, you wished to store an answer in a field and specify a width larger than was needed (to allow for modification later on, etc.). It is specified with a period, followed by the width. Different question types have different maximum widths (CAT=20, FLD=3,000, NUM=16, VAR=79 for terminal mode and 3,000 for other modes, TEX=1) or default widths. You may specify a width without specifying a specific location, such that the "next" location would be used with the width specified. If you specify a width that is too short for the question's requirements, an error message will be displayed when you compile the question.

You can specify a width greater than needed for any question. CAT and FLD questions will generate warnings.

```
EX:
{ [Counter.4]
```

This would put the data in the location of the previous question Counter with a width of 4.

```
EX:
    {  .2
```

This question would use the next data location and use a width of 2. If you don't assign locations, and have already started interviewing, you may want to save your locations to avoid having them shift if you add or delete questions later. You can do this by using the HARD_CODE compiler command to *hardcode* locations in the backup specification file (QSP) when you compile. Using this file to make future changes will ensure that you are warned if you overwrite columns.

If you do not use the HARD_CODE option, and wish to insert questions after interviewing has started without affecting unassigned locations, use the SAVE_COLUMN and RESTORE_COLUMN compiler command. (This would not affect subsequent questions.)

To delete questions and save the old location(s), use the HIDE option to hold the data space or set a specific column on the next question. If data is accidentally shifted for some interviews, you may be able to use Mentor or some other data manipulation program to shift it back later.

See *2.9.1 WHAT TO DO WHEN A STUDY HAS BEEN SOFT CODED* for more information on making changes to a questionnaire once interviewing has started.

If you specify a data location that has already been used by another question (either assigned by you or the program) or by the case ID, a warning message will be issued during compilation unless you use the compiler command -CHECK_COLUMN_OVERLAP.

Question Options

The question label line includes options to hide the question and saving the answer to the question under a previous alias name. If you use both options on the same question label line, HIDE must be before ALIAS.

The option HIDE hides the question. This lets you create questions for internal purposes that will not be displayed on the interview screen. You will want to use a question label or number on a hidden question to reference it with. The hide question is used as a position marker for subsequent IF conditions or expression (EXP) statements, or to create variables for later reports. You cannot display text on a hidden question's codes, but you can display the data. You can reference it in logic conditions or expressions.

The hide question uses up data space. However it will never execute or store anything under the ALIAS=name.

```
EX:
{Savedata: HIDE
```

This would create a hidden question called Savedata.

The alias question is the name of a previous question to store the answer to this question under for back-referencing purposes. The syntax is ALIAS=label. This tells the program to save the answer as if it were the response to the previous question specified. The data is not affected, just the answer array.

For example, you might have the same question in two different branches of the questionnaire and later you might want to back-reference the response to that question, no matter where it was asked.

If you alias the second version of the question to the first, your back-reference can just point to the first question's label and you will automatically get the respondent's answer no matter where the question was asked.

Sometimes people use ALIAS when they have an Other (specify) situation.

For example, if a question with a response list (i.e., CAT or FLD type) allows other as a response, and you have a later question (i.e., TEX type) asking specifically "What other?", the response to the TEX question can be saved as if it were a response to the original CAT or FLD question. Then, when the original CAT response is referenced in the display of a subsequent question, and the respondent has given some other response, this response would be displayed.

```
EX:
Other: ALIAS=Catques
```

This would store the answer to Other as if it was a response to Catques. Back-references to Catques would pick up this response if there was one, otherwise the response to Catques. Back-references to Other would pick up only the answer to the later question.

(See *Appendix Z: PRACTICAL APPLICATIONS* for an example). You can also use a popup question. (See *2.4.2 CATEGORY QUESTION RESPONSE LIST, Collecting Other (Specify)* Answers.)

Here are some sample question label lines with explanations:

```
EX:
{
```

No label, question number is flexible, next data location used, no options.

Label is ADDRESS, question number is flexible, data location is record one, column 23; no width is specified (width to be determined by PREPARE); and there are no options.

```
EX:
{DATE: 24.2
```

Label is DATE, data location is column 24, width is 2 columns, no options.

```
EX:
{ .3 HIDE
```

No label, use next data location with a width of 3, and the question will not be executed.

```
EX:
{RATINGS: [RATE1]
```

Label is RATINGS. The data location is the same as the prior question RATE1.

```
EX:
{EDUCATE: ALIAS=YEARS
```

Label is EDUCATE, and if asked, the response will be stored as if it was a response to the previous question labeled YEARS.

2.4 DATA ENTRY QUESTION TYPES

The data entry question types are used to collect data from the respondent or interviewer. The other question types are discussed in *Chapter 3: ADVANCED PREPARE SPECIFICATIONS*. Data entry questions display the text and, where appropriate, the possible response list on the screen along with the standard data entry prompt (-->).

The interviewer enters a response and **Enter** (or **ESC**, on certain questions and certain hardware platforms), except when using the AUTO_RETURN compiler command in which case you will see this prompt, ==> (DOS only), and the **Enter** is *not* required.

The data entry question types (and their common abbreviations) are:

- **Category (CAT).** This type consists of question text and a list of pre-defined responses. They may be single or multiple response. Each response places a binary punch (1, 2, 3, 4, 5, 6, 7, 8, 9, 0, X, or Y) in the data, which may be different than the response code. The maximum data width is 240 and the maximum number of responses is 240. *Examples:* Reversed Rating scales, products used.
- **Field (FLD).** Field questions also consist of question text and pre-defined responses. However, with FLD questions the actual response (rather than a punch) is placed in the data. They may be single or multiple response. In addition, disk-based FLD questions allow multiple screens and up to 10,000 return codes. The maximum data width is 3,000. *Examples:* State codes (CA,NY,TX), Rating scales, Products used, ZIP codes.
- **Numeric (NUM).** These are used when the response can be a number within a range. The numeric value of the response is placed in the data. The response is controlled by a range and a few exception codes. The maximum data width is 16. *Examples:* Number in household, miles driven, price.
- **Text (TEX).** Text questions are for long open-ended responses—or for extremely variable length open-ended responses. Up to 5,000 characters may be entered per TEX question. You can control the size of the response area (i.e.,

you can define a box for the answer that will allow fewer characters). The actual response is placed in the data at the end of the data file in compressed format. A full-screen editor is used for data input on monitors; terminals use a line editor. *Examples:* General likes/dislikes, other products.

- **Variable (VAR).** Variable questions are for short, controlled length, open-ended responses. The actual response is placed in the data and it can be from 1 to 3000 characters long. (Note: VAR statements may only be 76 columns wide in terminal mode). You control the minimum and maximum number of characters to be typed. *Examples:* Name, address.

Questions can span multiple screens if necessary. You will see "Press **Return** to continue" at the bottom of a screen to be continued. Read the appropriate section for more information.

The data entry question type, subtype, and options are on the line following the question text. The question type tells Survent what kind of data to expect as a response, whether there is a response list and how to prompt for a response.

The basic syntax for the question type is:.

```
!question type,subtype,options
```

Only the exclamation point and question type are required. The specific syntax varies for each type of question or control statement and is described for each data entry type below and in *Chapter 3* under control statements.

2.4.1 Category Question Type

With Category, or CAT, questions the respondent picks one (or more) of the answers from a list of coded responses or categories. CAT questions are useful for rating scales (Excellent, Good, Fair, Average, Poor, No Answer), lists of predetermined responses (such as Male, Female, No Answer) or longer lists.

A CAT question has a maximum data width of 240, and allows up to 240 responses.

CAT questions display the question text first, then the response list. If the response list is longer than the number of lines available on the screen, it is wrapped for as many columns as it takes to fit all of the responses on the screen, unless you use the compiler directive "Response_List_Columns=#" to force the program to use a specific number of columns. If the responses will not all fit on the screen, you can control what happens. See *HOW BIG CAN IT BE?* later in this section. The data entry prompt

is placed below the response list unless placed elsewhere with the _ parameter (see *2.5.1 SCREEN FORMAT CONTROLS, Positioning the Data Entry Prompt*).

CAT questions are functionally the same as FLD questions, except that they store the data using binary "punch" codes instead of using the actual codes typed by the respondent. This allows you to store more codes in less space, or "recode" data for other purposes. CAT questions, unlike FLD questions, do not blank data in their response location prior to being executed, so answers to more than one question may be saved in the same location.

The syntax for a CAT question type is:

```
!CAT,subtype(s),number of responses allowed,additional
parameters
```

The subtype, number of responses allowed, and additional parameters are optional. A comma is used before each additional parameter. The default is one allowable response and no subtype.

Advanced users note: Here, as in many other places, you're allowed to use a space instead of a comma when specifying several items in a row. A comma must be used, however, when accepting a default value and needing to specify a subsequent option.

The CAT question subtypes follow:

A applies response text to a code in the data without presenting a question to the

interviewer (i.e., the question is *hidden* from the interviewer). The CAT,A may be placed on a prior question to get the data it needs, or the data may be copied to it by a previous GEN, SPC, or PHONE statement. The response text corresponding to the filled codes in the CAT,A response list can then be displayed in subsequent questions. If there is no data for the CAT,A, or there are more responses than the maximum allowable, the Survent interview will terminate at the question with error #108 (see *Appendix D: SURVENT BLOW ERROR MESSAGES* for Survent error messages).

B allows the question to be unanswered (and thus blank in the data). Use this with caution as it is difficult to track the interviewers' movements if some questions are blank. (Subtype B is particularly useful in WebSurvent grids, when you want to allow blanks in some fields, but require answers in others. With a !CAT,B, no data will be recorded, so nothing will be in the answer array. (For more information on webSurvent grids, refer to the WebSurvent manual.)

For !CAT questions, you can combine subtypes "A" and "B" to have an auto-fill response that also allows blanks. (Users previously had to be careful to write conditions on the statement to keep it from executing when it is blank, or else it would get a blow #108 and the interview would terminate.) Use this with caution because no data is recorded and it may be hard to determine whether this is due to logic skips or skipping through the question.

N hides the response list during the interview. This is useful when a series of questions use

the same response list and you only want to display it once, or if you want to format the

screen differently from the default response list format.

C displays only those responses that the respondent has *not* selected in a prior question. All responses (previous and new) are left in the data. This subtype must operate on data generated from a previous question. See subtype “I” for an easier way to do this.

D displays only those responses that the respondent *has* selected in a prior question. The data is blanked of any responses that are not re-entered. This subtype must also operate on data generated from a previous question. See subtype “I” for an easier way to do this.

I allows you to include or exclude answers given to previous questions, or include/exclude particular responses in a standard list, without having to use GEN statements to initially define the data. Typically used for unaided/aided groups of questions with the same response list. See the FLD,I question for syntax.

Note: For CAT subtypes C, D, and I , you may want to display all responses in the response list, but mark (with asterisks) those that are not allowed as responses. You can do this by including the FLAG_DISALLOWED_CATS compiler command. For examples of CAT subtypes A, C, D, and I see *Appendix Z: PRACTICAL APPLICATIONS*.

M When using Highlightcats mode to mark responses on the screen, this causes the interviewer's text to match the TEXT of the response rather than the default which is to match the response code. This is especially useful for very long alphabetically sorted lists. You may also use the header statement HCAT_MATCH_TEXT" to have the questionnaire use this feature whenever Highlightcats mode is used.

R rotates (or "scrambles") the response list before displaying it, so the responses can be read to each respondent in a different order. All responses will be rotated, except for marked responses (see section 2.4.2) or comment codes.

If you have a complex rotation control, or you need to match the rotation specified on a group of questions, use the !ROTATE syntax in the code list as discussed under *ROTATING CODE LISTS*.

You may make the rotation order match a prior CAT question using the "rotate_seed=<label or location>" parameter. The <label or location> must be a 10 character field similar to the seed for ROTATE,S statements; if it is a label it must be prior to the current question. If the seed is blank the question will fill it with a new seed, if it has a value the question will rotate in the order specified by the seed. The questions to be rotated must have the same number of response items to rotate in the same order.

The ROTATE_SEED parameter must be the last one on the question; if it is after a subtype I included/exclude list, the list must end with ";EIE,".

S displays the response list in alphabetical order, sorted by response code.

T displays the response list in alphabetical order, sorted by response text.

X allows you to re-enter responses and not overwrite existing codes. See *ADDING CODES* below for more information.

Combinations of subtypes may be specified by entering the two codes in a row.

```
EX:
!CAT,XR
```

Options R, S, T, N and A cannot be used together.

The number of responses allowed can be any number from one to the number of responses in the response list. If more than one response is allowed, certain responses can be marked as being "exclusive." That is, if they are entered, no others may be entered (see *EXCLUSIVE RESPONSES* in the next section).

Here are some example CAT question type lines:

```
EX:
!CAT
!CAT,R
!CAT,,5
!CAT,R,3,ROTATE_SEED=SAVEROT
!CAT,I,3,IL=Q3;XR=07
!CAT,IR,3,IL=Q3;XR=07;EIE,ROTATE_SEED=SAVEROT
```

How Big Can It Be?

By default, if a CAT or FLD question is bigger than one screen, it will display the first screen. Then it will wait for you to press **Enter** to display the next screen, etc. In this mode, there is no ability to toggle back and forth between screens. Specifying the !Highlightcats directive allows toggle capability. Having questions bigger than a few screens may require the use of the header option MAXIMUM_QUESTION_SIZE=64000. The default is 32,000. The maximum number of codes you can use depends on how much text is on the codes, but it would be on the order of 20 screens of codes.

(This means you can have multi-response questions with up to 800 codes (20 screens of 30 codes), or 2,000 codes with no text.)

Highlightcats mode allows you to use multiple page response lists. If you have more than one page full of responses, the text for the first screen will display. Then you can move from page to page by entering codes on a particular page or using **PageUp** and **PageDown** (or **Ctrl-F-U** and **Ctrl-F-D** on terminals).

If you type a response code, it will take you to the page the response is on and highlight the response there. If you mark the item as a response, go to another page, then return to the original page, the highlighting will still be there.

To keep the question text on the screen between pages, use a box specification such as \(5) at the bottom of the question text to keep the response list below that position.

Remember, there is also an overall limit of 240 responses for CAT questions. FLD questions allow 1.000 responses. For lists with more response codes allowed but only one response, see *3.1.5 DISKBASED* FIELD QUESTIONS FOR LARGE RESPONSE LISTS.

See *3.2.1 INTERVIEW CONTROL COMMANDS* for more information on Highlightcats.

Adding Codes

CAT,X OR FLD,X allow you to re-enter responses to a CAT or FLD questions and not overwrite existing codes. You may use use "+code" or "-code" to add or subtract codes. It also displays any codes that were already in the data. The +code or -code string can combine codes such as +123-45 to add codes 1,2, and 3 and take out codes 4 and 5. This is most useful for after-the-fact Codingmode questionnaires, where you are coding open-end responses into existing code lists. In Highlightcats and Echocats mode, the items already in the data are marked as chosen already when the question is presented. For these modes, it will highlight the responses already chosen and you can then add or subtract codes.

To add or subtract codes, the second question must use the same data location as the first. In regular mode it displays (010406)-> in the prompt to show that codes 01, 04, and 06 are already chosen; you can then say "-01+05" etc. to subtract the “01” and add an “05 code.

Note: This subtype allows you to leave the question without making any changes, even if the question is initially blank. If this is a concern, put a question after the CAT,X to reset if the location is blank.

2.4.2 Category Question Response List

After the question type line, you must specify the response list; this is the list of possible responses and the codes the interviewer will enter for each. The maximum number of response list items is explained at the beginning of *2.4.1 CATEGORY QUESTION TYPE.* The response list also allows you to tell the program to skip to another question if a certain response is selected, alter the punches which will be stored in the data, mark a response as being exclusive in a multiple-response question, or designate a question to pop-up on the screen immediately to allow the interviewer to capture something such as an other (specify).

The syntax for a response list item is:

```
 (column offset,-,!,punch,SKIPTO label,
O=pop-up, other options) response code text
```

Parentheses are not required unless one of the optional parameters within the parentheses is used.

If specified, the response list options must be entered in the order shown; a comma placeholder is not required for unused parameters. The only required item on the line is the response code that the interviewer will type as a response, but text is also usually supplied to delineate the reason for the code. The response code is limited to nine characters. Here is a typical response list:

```
EX:
A Apples
B Oranges
C Bananas
D Other fruit
```

As many spaces as you want can be placed between the code and the text, but only one space will be shown during the interview unless you use a caret (^) as a place holder. See the *RESPONSE TEXT* section later in this chapter for more information. Specify one response list item per line, using as many lines as you have items. When the question is presented to the interviewer, the response list will be wrapped into multiple columns to fit the available space.

The column offset and punch refer to the data stored for a particular response. Since these will be automatically assigned by the program if not specified, you will usually not need these. The dash indicates an exclusive response. The exclamation point keeps that item fixed when the list is rotated or sorted. SKIPTO label tells the program to skip to some other question if this response is chosen. The pop-up designates the question to immediately pop-up on the screen when that answer is chosen. More details on each of these are given below.

Response List Options

The column offset and punch cannot refer to more than 240 columns. By default, PREPARE will use the current column, punches 1 through Y for the first 12 categories, then go to the next column and use 1 through Y for the next 12, etc. If you specify an offset or new punch, the next item will be given the following punch in the same column. You may want to specify that the punch for a response be placed in a column other than the default.

- To place the data in a particular column of the question, use the +# syntax, where # is the relative position of the column from the start of the question.
- To place the punch in the second column of the question, enter +1 as the offset.
- ·To place the data back in the first column of the question, enter +0 as the offset.
- The column offset +# additionally requires that you specify the punch you want to use.

You may also use the syntax ++# for the column offset, to place the punch in a column relative to the current column, that is, the column of the last response item's punch. This is typically used to just move ahead to the next column before you have used all the punches for a particular column. To place the data in the first column beyond the previous item's column, enter ++1 as the offset. ++# does not require a punch, and defaults to the 1 punch of the next column.

The next response in the list will always go in the same column as the last response, (or in the next column if a Y punch is reached), unless another offset is specified.

You may also wish to specify the punch to store in the data for a response. The punch may be 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, X, or Y.

```
EX:
(++1,Y) A Apples
```

This would store a Y punch in the next column if a response of A is entered.

A dash (-) makes the response exclusive—if that response is chosen, no other responses are allowed. This can be useful, for example, with a Don't Know response in a multiple-response question. The dash by itself would use the next available punch, but mark the response as exclusive. If also specifying an exclamation point, the comma between the dash and the exclamation point is optional.

```
EX:
(-) ZZ Don't know
```

This would make the response ZZ exclusive; no other responses could be entered with it.

An exclamation point (!) keeps the response in its absolute position when using a subtype that rotates or sorts the list (see subtypes R, S, and T).

Punch codes save data space but do not keep track of response order; if you want to record the responses in response order, either use FLD questions instead of CAT questions, or recode the punch data to be response codes (like FLD questions) by using LOOP/ENDLOOP statements controlled by the original punched CAT question; the responses will be generated in response order.

Autopunches Mode

Autopunches mode controls whole blocks of questions ({!AUTO_PUNCHES} and {!-AUTO_PUNCHES}) or a single question (!CAT*). Column offset and punch are not allowed with either. Using these, the punch stored is controlled by the response code. A response code of 2 puts a 2 punch in the first column, a 12 response is the Y punch of the first column, 14 is the second punch of the second column, etc.

Autopunches mode allows response codes 01-99 (for one or more columns), 001-240 (if you're spreading the data over many columns) or 1-0, X, and Y only. You may still mark a response as being exclusive. (See *3.2.2 COMPOSE CONTROL COMMANDS* for more information on this mode.)

Skipping Forward or Backward Using Response Codes

If you want the program to skip to another question when a certain response is given, specify that with SKIPTO label or SKIP label. The label specifies which question the program should skip to if the response is chosen. If the sending question is multiple-response and the responses dictate different skip patterns, the interview will skip to the closest place.

```
      EX:
(SKIPTO DEMOGRAF) 03 Don't like any cars seen so far
```

This would skip to the question called DEMOGRAF if the 03 code was chosen.

Warning: Skipping backward in a questionnaire does not remove previous data. Use only after careful testing!

Pop-up Questions - Collecting Other (Specify) Answers in Response Lists (O=label); POPUP= also an option

If you have a category such as Other (specify), where you do not have that answer listed, but want to give the interviewer a chance to type in the verbatim before they forget it, you can designate a question to "pop-up" on the screen once that category is chosen. The interviewer would then type in the verbatim, press **Enter** or **ESC** to exit that question, and be placed back on the original question, ready to enter additional responses.

To designate the question that will pop-up, enter O= followed by the question label or number of the question.

```
EX:
(O=qlabel) 0 Other
```

You must be in Highlightcats or Echocats mode. In Highlightcats mode, you will go to the designated question as soon as you have chosen the marked response, and then return to the original question with the existing codes intact (including the “other”) to enter more codes. In echocats mode it will ask for the response after all other responses are entered. The question specified on the O= reference must be defined prior to the question that calls it. It will not be executed until it is called, so you do not have to hide the question or put an IF condition on it.

To bring up the Other question on the same screen as the original, define the Other question’s text in a non-conflicting screen location (eg. \(15,20,15,40))so there is no overwriting of other text on the screen.

Here is a sample setup of that portion of a questionnaire:

```
{!HIGHLIGHTCATS}
{ Other1:
```

```
\(10,40,15,77)
Enter the Other code now

!VAR,,20 }
{ Q13:
What products were advertised during the Olympics
coverage?
!FLD,, 4
1 Budweiser
2 Coca Cola
3 McDonalds
(O=Other1) 4 Other: \:Other1:^20
}
```

Note the use of ^20 in the above example to control the width at which the VAR response prints. “-DISPLAY” and Other Response Code Options Other options are used to make it easier to automatically generate codes for CfMC’s Mentor tabulation package or others. One of these is “-DISPLAY”, will include the code in the variable created but will not display it on the screen. This can then be used as a code for tables or coding later. See the MENTOR manual for other useful keywords for automatic table generation.

“Popup=” is another option to 0=

!FIELD and !CATEGORY questions support "popup=" in addition to "o=" syntax for pop-up question references. You can type "POPUP=Q1O", for instance, instead of "O=Q1O" when you specify the question that you want to pop up. For instance:

Example:

```
{Q10: !text,,1}
{Q2:
Which of those is your favorite
!FLD
1 Apples
2 Oranges
(Popup=Q10)
3 Other fruit }
```

Positioning the Response List

You can position the response list independent from the question text by using a screenpositioning box (\(#,#,#,#)) at the end of the question text:

```
EX: {
What is your occupation?
\(10,10)
!CAT
1 Butcher
2 Baker
3 Candlestick Maker}
```

This would display the question text on the first line of the screen, but the response list starting on the tenth line and tenth column of the screen. (See *2.5.1 SCREEN FORMAT CONTROLS, Using Boxes to Specify The Screen Position,* for more details.)

Displaying a Response Code Delimiter

A special option in response list positioning is the specification of a delimiter character to separate the response codes from the text. Place that character immediately after the box specification.

```
EX:
\(5,5))
```

This will place a ")" after each response code on the screen as follows:

1) Butcher

2) Baker

3) Candlestick Maker

You may use any single ASCII character; the interviewer will enter the simple response code (i.e., 1, 2, or 3 in the above example) as before.

Response Codes

The response code is the character set that the interviewer will type as a response. These codes can be from one to nine characters and are usually all the same length (see exception below). Response codes may be any alphanumeric characters, plus a period or underscore. Each response code must be unique within the set used for one question. The response code is followed by one or more spaces, and then the text of the response.

A caret (^) may be used at the end of a response code as a placeholder in cases where you want different lengths for your response codes.

```
EX:
YES Yes, I like it
NO^ No, I don't
```

This would allow the responses YES or NO (case-insensitive), and nothing else. If you use all carets as a response, you will allow the interviewer to press **Enter** as a response. This will still store a punch for CAT questions, but will not store anything in the data for FLD questions in the data. This item usually has no text because it is just being used to allow **Enter**.

```
EX:
YES I like it
NO^ Don't like it
^^^
```

If **Enter** was pressed by the interviewer, a 3 punch would be stored for a CAT question.

Comment-only (Equal Sign) Response Codes

The equal sign (=) response code indicates that this response cannot be chosen by the interviewer. Use this option to format or annotate the screen presented to the interviewer. In Highlightcats mode, the highlight bar will be positioned on an equal sign response code if it is the first response, therefore forcing the interviewer to do more than press **Enter** to move from this question on single response questions.

- Equal sign response codes must be the same length as other response codes
- Equal sign responses will not move in a rotated list, although things can move around them
- Parentheses may not start the text unless except when used with a list of valid response codes to generate a 'net' category for the DEF file passed to Mentor
- Equal sign responses do *not* get a punch in the data
- You can specify any number of equal sign responses in a single question; PREPARE does not count them as duplicate responses.

NOTE: You can use the equal sign code to keep a blank line between rotated codes (CAT,R and FLD,R) and a Don't Know response. Another use is to put a heading inside the list.

```
EX: {BRANDS:
Which brands?
!CAT,R,3
01 BRAND ONE (only responses 01-04 will rotate)
02 BRAND TWO
03 BRAND THREE
04 BRAND FOUR
== (displays a blank line that does not rotate)
(-,!) 05 DON'T KNOW }

( ! means this response will not rotate)
```

Response Text

The response text can be as long as will fit on one line (up to the number of characters indicated on the Header option SPEC_WIDTH) or can be continued onto another line by ending the first line with an ampersand (&) and continuing the text on the next line (with a maximum of 5000 characters).

Continued lines do not need to be indented in your spec file; Survent will indent text properly on the screen. Text is not required—you could use just a response code. As with question text, you may make text enhancements or refer to other answers using a backslash (\) and special text control characters (see *2.5.1 SCREEN FORMAT CONTROLS* and *2.5.2 TEXT ENHANCEMENTS AND GRAPHIC CHARACTERS)*. You may insert a line break in the response list by using backslash N (\N).

To indent the response text, use leading carets (^). The caret symbol will only appear in your specification file or in the Script Composer - not on the interviewer's screen.

```
EX:
{QCAT:
What type of fruit is your favorite?
!CAT
1 Red fruits (general):
2 ^^^Apples
```

```
What type of fruit is your favorite?
1 Red fruits (general):
2 Apples
-->
```

Here is a sample CAT question and response list and the corresponding interviewer screen:

```
EX:
{ATT:
How often do you use AT&T to make long distance
calls?
!CAT
1 More than once a week
2 2 - 4 times per month
3 About once a month
4 Less than once a month/\NNever }
```

```
How often do you use AT&T to make long distance calls?
1 More than once a week
2 2 - 4 times per month
3 About once a week
4 Less than once a month/never

-->
```

The question label is ATT. The question type is CAT, no subtype is specified, and only one response

is allowed (because no number is specified). The last response item will print on two lines because of

the \N before the word Never. The end of question marker (}) can be after the last item's text, or on the next line.

NOTE: See *2.5 QUESTION TEXT* for other backslash codes that can be used for question text or response list text.

Response List Blocks for Languages

You may group response items for purposes of language definition. The blocks start with a **[** and end with a **]**. Here is a language block:

```
EX:
[L=EN]
1 yes
2 no
[L=SP]
1 Si
2 Non
etc.
```

This syntax makes it much easier to define languages than using \L commands to separate the text on each response, especially for long lists or when referring to items using &file references (see 2.5.5 SPECIFYING MULTIPLE LANGUAGES).

Using the Same Response List Multiple Times

You may want to use the same response list for more than one question. Simply type [SAMEAS label] in place of a response list and the program will use the one from the specified question. This SAMEAS option can only be used if the question exists prior to the current question and the two questions are the same question type. [SAMEAS label] will cause not only the same response codes and text to be used, but also the same punches, exclusive mentions, and skip patterns (if any).

The number of responses allowed must be re-specified on the question type line.

```
EX:
{MCI:
How often do you use MCI to make long distance
calls?
!CAT
[SAMEAS ATT] }
```

In this example, the response list from the question labeled ATT (in our previous example), including

the SKIPTO with response 5, will be used.

In addition to Autopunches mode described earlier, there are other special modes for CAT questions—Highlightcats mode and Echocats mode. Also see *3.2.1 INTERVIEW CONTROL COMMANDS* for more information on interviewing modes.

Rotating Code Lists

There are two ways to rotate code lists. For a simple rotation of a code list, you can use subtype “R” to invoke the rotation. This will rotate the list in a scrambled fashion, except for codes that are comment codes (eg. == codes) or marked to not rotate (eg. (!)). You can also force a code list to rotate the same as another code list using the ROTATE_SEED option on the question type line.

However, if you have other complexities, or want to match a set of questions previously rotated, you will need to mark the code list the same way a set of questions is marked (see the !ROTATE compiler command). You can do this on the code list by using !ROTATE syntax and inserting !FIX or !GROUP statements wherever necessary to invoke the rotation control you want.

You may only have one !ROTATE statement and it must be specified at the top of the code list. For example:

```
EX:
{Q23:
What type of car do you have?
!FIELD
!ROTATE,S,1,[1006.10]
1 Audi
!GROUP,3
= Chevy:
2 Truck
3 Camaro
!GROUP,3
= Ford:
4 Mustang
5 Truck
!FIX
9 Other
0 Don't know
!END_ROTATE}
```

This will rotate the question and store the rotate seed in 1006.10. You can use the rotate seed to control other questions so that your list and set of questions rotate in the same order. You can still use the !FLD/CAT,R statement to rotate a list and the (!) glyph to fix the position of an item on that list, but that syntax doesn't save a seed you can use with other question sets and does not have the !GROUP/!FIX controls; it also will only allow “scramble” rotates while the !ROTATE syntax allows all three rotate types (scramble, regular, and flip).

Highlightcats Mode

If you use the HIGHLIGHT_CATS compiler command prior to a CAT or FLD question, the response list will be displayed with the first response highlighted. You can use the arrow keys (on a monitor; use **Ctrl-U**,**D**,**R**,**L** on a terminal) to move around on the list or you can type a response code and the highlight bar will move to that position on the list. **ESC** or **Enter** will accept the response for single response questions (+/- or Ins/Del for multiple response questions). (See *3.2.1 INTERVIEW CONTROL COMMANDS*, *{!HIGHLIGHT_CATS)* for more information on this command.)

NOTE:

- Questions in this mode should *not* use the caret (^) as a response code (see *2.4.2 CATEGORY QUESTION RESPONSE LIST, Response Codes*). When the interviewer presses **Enter**, the program will back up to the previous question instead of entering the appropriate punch as a response.
- The study header option ECHO_CATS overrides Highlightcats mode.
- Using Highlightcats mode will also cause questions that are too big to fit on one screen to display on multiple pages.

Echocats Mode

If you use the ECHO_CATS header option, the ECHO_CATS compiler command, or have Echocats capability assigned to you in the employee file, you will be able to enter multiple responses to CAT or FLD questions one at a time, pressing **Enter** in-between responses. Each time you press **Enter**, the chosen response(s) will have its (their) response code(s) highlighted so you can more easily identify the responses already entered. This happens only on multiple-response questions unless the ECHO_CATS option specifies to do it on single-response questions as well.

2.4.3 Field Question Type

Field, or FLD, questions are like CAT questions in that they use a predetermined list of responses. With FLD questions, however, the interviewers' entries are not recoded to punches — the program saves the answer exactly as entered. FLD questions allow single or multiple responses per question.

PREPARE will assign sufficient data columns to accommodate the maximum number of responses allowed. The default is no subtype, and one response. Existing data in the location of the FLD question will be overwritten when it executes.

FLD questions have a maximum data width of 240. There is also an overall limit of 700 codes per FLD question. See the discussion under the CAT question, *HOW BIG CAN IT BE?*, for details on questions too big for one screen.

The responses are stored in the order entered. All answers are stored in consecutive columns.

The syntax for a FLD question type is:

```
!FLD,subtype(s),number of responses
```

The FLD question subtypes are similar to the CAT subtypes, and they are:

A applies response text to a code in the data without presenting a question to the interviewer (i.e., the question is *hidden* from the interviewer). The FLD,A may be placed on a prior question to get the data it needs, or the data may be copied to it by a previous GEN, SPC, or PHONE statement. The response text corresponding to the filled codes in the FLD,A response list can then be displayed in subsequent questions. You will be aborted from the Survent interview with error #108 if this executes and there is no data

in the field matching a response code unless you have used the caret (^^) response code as one of your codes to allow blank. FLD,A response lists may include a catch-all category at the end consisting of question marks (eg. ?? text), where there are as many question marks as needed to match the number of characters in the response codes of other items. This category will then allow unmatched items in the data and assign them the corresponding text... (See *3.1.4 DATA ENTRY QUESTION FUNCTION MODIFYING STATEMENTS* for an example using this catchall category.)

B allows the question to be unanswered (and thus blank in the data). Use this with caution as it is difficult to track the interviewers movements if some questions are blank.

(This is useful in webSurvent grids, when you want to allow blanks in some fields, but require answers in others. With a !FLD,B, no data will be recorded, so nothing will be in the answer array. (For more information on WebSurvent grids, refer to the webSurvent/webCATI manual.)

For !FLD questions, you can combine subtypes "A" and "B" to have an auto-fill response that also allows blanks. ***NOTE:*** Use this with caution because no data is recorded and it may be hard to determine whether this is due to logic skips or skipping through the question.

I allows you to include or exclude responses from prior questions with the same code list or include/exclude particular codes in the list.

- Lets you include responses from some questions and exclude responses from others.
- Does not require moving data into the location first with a GEN statement.
- Records the new answers only.

Here is the syntax for a FLD question subtype I:

```
!FLD,I,# responses,&
IL=<question labels>; XL=<question labels>; &
IR=<response list>; XR=<response list><;EIE, ...>
```

IL (or INCLUDE_LABELS) tells which questions' answers to include as allowable for this question. By default all answers are allowed.

XL (or EXCLUDE_LABELS) tells which questions' answers to exclude as allowable for this question.

IR (or INCLUDE_RESPONSES) tells which responses to also allow, even if not mentioned previously.

XR (or EXCLUDE_RESPONSES) tells which responses to not allow, even if mentionedpreviously.

NOTE: !FIELD and !CATEGORY subtype I questions support a range of categories in the "INCLUDE_RESPONSE" and "EXCLUDE_RESPONSE" lists. Previously, the only supported codes were separated by commas. So, for instance, you can use:

Example

```
      !FIELD,I,,IL=LASTQ;IR=01-05,23,44,91-99;XR=87-89
```

J only shows response codes with text.

This is designed so that you can have a question with back-references on it, and only the ones that have been filled will be displayed. This is similar to the !FIELD,I,IL= (include label) feature, except that you don't have to have a controlling label filled in order to determine what to do on this question.

Example:

```
!FLD,J
      1 This always shows
      2 \:Q1: ''This might not show
      3 ''This will never show
 }
```

;EIE, Ends the include/exclude list so that other parameters may follow; not used unless there are other parameters such as ROTATE_SEED.

NOTE:

- Specify options in order shown; while not required, they indicate the precedence given (XR over IR over XL over IL).
- If there is no INCLUDE question, the default is to include all responses on the list.
- Subtype I can be specified in addition to subtypes N to hide the list, R to rotate the list, or S or T to order by response code or text.

- The lists from question to question don’t have to match but must have the same response code width. Unmatched codes from list items are ignored.

You can now control which codes will be displayed in webCATI and terminal mode but not in webSurvent. To do this, use the !FLD,I subtype with the statement WSXR=<list of codes>, for example:

```
     EX:
{Q1:
dont show  DK to websurvent
!fld,i,,wsxr=3
 1 yes
2 no
3 dk
}
```

M When using Highlightcats mode to mark responses on the screen, this causes the interviewer’s text to match the TEXT of the response rather than the default, which is to match the response code. This is especially useful for very long, alphabetically-sorted lists. You may also use the header statement HCAT_MATCH_TEXT" to have the questionnaire use this feature whenever Highlightcats mode is used.

N hides the response list during the interview. This is used when a series of questions uses the same response list or if you want to format the screen differently from the default response list format.

R rotates (or scrambles) the response list before displaying it, so the responses can be read to each respondent in a different order. All responses will be rotated except for exclusive responses and equal sign responses (see *2.4.2 CATEGORY QUESTION RESPONSE LIST*).

If you have a complex rotation control, or you need to match the rotation specified on a group of questions, use the !ROTATE syntax in the code list as discussed under *ROTATING CODE LISTS*.

You may make the rotation order match a prior FLD question using the "rotate_seed=<label or location>" parameter. The <label or location> must be a 10-character field similar to the seed for ROTATE,S statements; if it is a label it must be prior to the current question. If the seed is blank the question will fill it with a new seed, if it has a value the question will rotate in the order specified by the seed. The questions to be rotated must have the same number of response items to rotate in the same order.

The ROTATE_SEED parameter must be the last one on the question; if it is after a subtype I included/exclude list, the list must end with ";EIE,".

S displays the response list in alphabetical order, sorted by response code.

T displays the response list in alphabetical order, sorted by response text.

X allows you to re-enter responses and not overwrite existing codes. See *ADDING CODES* in the CAT question section for more information. See *HOW BIG CAN IT BE?* in the CAT question section for information on multi-page questions.

Options A, N, R, S, and T cannot be used or combined with eachother.

To combine codes, specify them immediately after each other in any order.

```
EX:
!FLD, SZ,3
```

Since FLD questions store the response code, the default width is the width of the response codes times the maximum number of responses. The response code can be up to nine characters. There is an overall limit of 240 columns.

2.4.4 Field Question Response List

As with CAT questions, after the question type line, you must enter the response list — which includes all the possible responses and the codes for each. You may also direct the program to skip to another question if a certain response is selected or make a response exclusive with a dash (see *2.4.2 CATEGORY QUESTION RESPONSE LIST, Response List Options)*. You can also designate a question to pop up if that response is chosen. Because FLD responses are not recoded, you do not specify column offsets or punches.

The syntax for a response list item is:

```
(-,!,SKIPTO label, O=label) response code text
```

The parentheses are not required unless you are specifying one of the options. The text is also not required — you could have just a response code. The same rules and features apply to FLD question response codes and text and skipping to another question as for CAT questions earlier.

The codes may be letters and/or numbers; all the codes must be the same length (see caret exception for CAT question) and there is a maximum of nine characters. Some special characters are allowed as codes (such as period and underscore), but should be used only in special cases and should be tested thoroughly. All alphabetic codes are changed to capital letters when stored in the data (i.e., abc becomes ABC). As with the CAT, each response code for a particular question must be unique. You can also use [SAMEAS label] to use a previous response list.

As with the CAT question, you can use the caret (^) response code to allow the interviewer to press **Enter** as the only response. Unlike the CAT, the caret response for the FLD question does not put anything in the data.

Here are two sample FLD questions:

```
EX:
{HOME:
Which of the following describes your home?
!FLD
A Apartment
C Condo
d Duplex
H Home (single-family)
N None of the above }
```

In the above example, the first initial of each response is used for its code. The code is stored in the data as entered by the interviewer.

```
EX:
{STATE:
What states have you visited?
!FLD,,3
CA California
OR Oregon
WA Washington
(-, SKIPTO OTHRSTAT) XX Other }
```

In the above example, the state code(s) would be stored in the data and a response of XX would cause a skip to question OTHRSTAT. The codes will be placed in the data in response order with no spaces in-between. The above example would reserve six columns in the data file – three possible responses times two character codes.

Highlightcats mode and Echocats mode are also available for FLD questions. (See *2.4.2 CATEGORY QUESTION RESPONSE LIST, Highlightcats Mode* and *3.2.1 INTERVIEW CONTROL COMMANDS*.) Also see *3.1.4 DATA ENTRY QUESTION FUNCTION MODIFYING STATEMENTS* for another variation on FLD questions, the disk-based response list for large or multi-level brand lists.

2.4.5 Numeric Question Type

Numeric, or NUM, questions are used to collect data such as price, miles driven, or number of people in a household. NUM questions have several options to let you specify various parameters of acceptable response.

The syntax for a NUM question type is:

```
!NUM,subtype,<NUMDECS=>#,range,other number,
exception codes
```

Specifying a subtype or any of the other items is optional. The data location width default is to the maximum needed for the allowable range, and can be up to 16 columns (a 10 digit number, preceded by a + or -, a decimal point, and four decimal places).

Using the NUMERIC_WIDTH_REQUIRED header option requires that you specify either a range, or a width on the question label line. The default width is 9 columns unless a range is specified. Following the order of the above syntax, commas must be used as placeholders when an item is not included if specifying a later item. Notice that range sometimes requires two comma placeholders (see below).

The NUM question subtypes are:

B allows the question to be unanswered (and thus blank in the data). Use this with caution as it is difficult to track the interviewers' movements if some questions are blank. (This subtype is useful in webSurvent grids, when you want to allow blanks in some fields, but require answers in others. With a !NUM,B, no data will be recorded, so nothing will be in the answer array. (For more information on webSurvent grids, refer to the webSurvent manual.)

R displays a rating scale line centered on zero, allowing a response +/- the maximum value specified by using the arrow keys (on a monitor; **Ctrl**-**R** or **L** on a terminal) to move to a position on the line. This subtype is described in more detail below.

V allows varying numbers of decimal significance, up to a maximum of the NUMDECS value.

X saves enough room for the response to include a plus or minus sign along with the number. This is necessary if converting the data to an SQL database later.

Z zero-fills the data field. The default is to right-justify the number, and precede it with blanks.

Number of decimals to use

The NUMDECS option tells the program how many decimal places of significance the interviewer must enter. If specified, the interviewer will have to enter a decimal point and the required number of digits (unless subtype V is used). The decimal point will use a column in the data. The default is no decimal places allowed (not even a decimal point). NUMDECS lets you specify a number from 1-8.

You may just specify the number without the keyword NUMDECS if you wish.

```
EX:
{
How much did you pay for the umbrella? (2
decimals, please)
!NUM,,NUMDECS=2}
```

This would require a response with 2 decimals like 2.23.

If subtype V is specified, the interviewer can enter a whole number or differing amounts of decimal significance. For example, with !NUM,V,NUMDECS=3 the interviewer could enter 5 or 5.2 or 5.23 or 5.234. All answers would be stored right-justified with the number of decimals specified in the numdecs= parameter.

Furthermore, if you want to save the result with no decimals, you can use the keyword NODECIMALS in the "numdecs=" place.

```
EX:
{
How much did you pay for the umbrella? (no
decimals, please)
!NUM,,NUMDECS=NODECIMALS}
```

NOTE: What is put into the data for a NUM question is the numeric value of the response, which is not necessarily the same as the number entered by the interviewer. If the interviewer enters 0025, either 25 or 25 preceded by zeros (if using subtype Z) or 25.00 (if subtype is V and numdecs=2) or +25.00 (if subtype is X) will be put into the data.

Range Description

Specifying a range lets you limit the range of allowable responses. The range may be specified with a few different syntaxes. The most common is to put the minimum value, a dash (-), and the maximum value. These values can be positive or negative numbers up to 16 digits including a positive or negative sign.

Supported numbers are in the range of - (2**31) to (2**31) or approximately +-2,147,000,000. Do not enter commas as part of the number if using large numbers. A comma can be used in place of the dash to separate the minimum and maximum values. If specifying NUMDECS, the range can include as many digits of significance as the NUMDECS allows.

If not specified, the default range is 1 to 999,999,999.

```
EX:
!NUM,,NUMDECS=1,1-10.5

EX:
{
What is the value of your assets?
!NUM,,,0-9999999}
```

This allows a numeric response from 0 to 9999999. The interviewer never has to enter leading zeros. You may also make either end of the allowable range relative to another question. In place of a number, you could enter a question label. The allowable minimum or maximum would be the previous question’s response.

```
EX:
{
How much of this money is used for household
expenses?
!NUM,,,1-TOTMONEY}
```

This allows a response from 1 to the response in the question TOTMONEY. The numeric range can also have data location references. Use brackets ([ ]) around the location(s).

```
EX:
{
Enter the amount of your salary you used for
vacations
!NUM,,,[10.5],[20.5]}
```

This says that the minimum is the number in column 10 for a length of 5 ([10.5]), and the maximum is the number in [20.5]. Make sure you have valid numbers in the locations if you use this type of reference. This example also shows using a comma instead of a dash in the range.

Numeric questions can also use <=# or >=# as the range.

```
EX:
{How many of the children are less than 10 years
old?
!NUM,,,<=TOTCHILD}
```

This says to allow responses less than or equal to the response to TOTCHILD. The less than and greater than characters (< and >) will be interpreted as <= and >=.

If you don’t specify the range so that the upper and lower ends are indicated, you'll need to leave an extra comma placeholder if anything else follows on the line.

Requires comma	No extra comma required
0	1-10
Nothing	[15.3], [21.3] > = TOTCHILD

“Nodecimals” can be used in the “numdecs=” place to indicate that you want the number saved with no decimals (decimals are truncated).

NOTE: If you do not specifically state the minimum value (low end) of the range, it will default to 0.

Using the Same Range Multiple Times

You may want to use the same range for more than one question. Simply type SAMEAS label in place of the range and the program will use all of the parameters from the specified question. This SAMEAS option can only be used if the question exists prior to the current question and the two questions are both NUMs. SAMEAS label will cause all of the previously specified parameters to be used, including: subtype, NUMDECS=, range, other number, and exception codes. A space or a comma can be between NUM and SAMEAS.

```
EX:
{PROD1:
How much would you pay for product 1 given a
price range of $1.99 - $3.99?
!NUM,,NUMDECS=2,1.99-3.99,,DK,RF}
{PROD2:
How much would you pay for product 2?
!NUM,SAMEAS PROD1}
```

Other Number and Exception Codes

If you specify a range, you may also want to allow some other number outside the range. Specify this after the range. The other number will be right-justified in the data field and will be affected by the Z subtype.

This other number must be a single number, not a range of numbers, or it can be a label of another NUM question.

```
EX:
!NUM,,,1-10,KIDS,DK
```

This example would allow a response of 1-10, the numeric answer to the question KIDS or DK as an exception code.

```
EX:
{
How many are used at home (1-50, or 99 if don't
know)
!NUM,,,1-50,99}
```

This would allow a response from 1-50 or 99.

Up to three other exception codes may be specified. They may use any alphanumeric characters, and most other characters (except ~, {, }, !, [,] and comma), and must be one to ten characters long. Upper and lowercase are treated the same; the answer goes into the data as uppercase. Exception codes are stored left-justified in the data rather than right-justified, even if they are numeric. Exception codes will not be affected by the Z subtype. Some standard exception codes are DK (Don’t Know), NA (No Answer) and RF (Refused).

```
EX:
{
What is your age? (RF for refused)
!NUM,,,18-99,,RF}
```

This allows responses from 18 to 99 or RF for refused.

NUM questions can have their exception codes go into the data as X or Y punches by specifying X* or Y* to put the punch equivalent in the data.

```
EX:
!NUM,,,1-9,,X*
```

In the example above, the range is 1-9 and the exception will be an X punch.

NOTE: The interviewer only enters an X or Y in this situation. If the data field is more than one column wide, the punch will be left-justified in the field.

If you use the caret (^) as an exception code, the interviewer can press **Enter** to move to the next question. The caret is left in the data as the response to the question.

If you use CR as an exception code, the question will allow **Enter** to continue. The characters CR are put into the data. Questions using this feature must be at least two characters wide.

NUM questions have no response lists. The range, other number and exception codes are the ways to limit the allowed response. Here are some other sample NUM questions:

```
EX:
!NUM,,,,,,DK}
```

Note that six commas are used as placeholders. The only specified item is the alpha exception code DK. The allowable range defaults to 0 through 999,999,999.

```
EX:
How many children do you have?
!NUM,,,0-20,,DK,NA,REFUSED }
```

No question subtype or number of decimals is specified. The range is 0 - 20. There is no other number specified and there are three alpha exception codes: DK, NA and REFUSED.

```
EX:
How many apples did you eat this month?
!NUM,Z,,0,FRUIT,99}
```

The subtype is Z. No decimals are allowed. The minimum is zero and the maximum is the response to question FRUIT. Another allowable number is 99.

Subtype R

This presents response boxes if there are 20 entries or less, or a number line for ratings with 21 to 79 entries. It only uses the maximum value out of the list of possible options to create the scale. It is most useful for self-administered questionnaires, where the problem of number bias is present.

Here is the display for 20 entries or less:

```
[ ] [ ] [ ] [ ] [ x ] [ ] [ ] [ ] [ ]
```

And here is the display for 21 entries or more:

```
|-------------------------*-------------------------|
```

The respondent is positioned in the middle and uses the left and right arrow keys (on monitors; **Ctrl**-**R** or **L** on terminals) to position themselves in the boxes or on the line. When they press **ESC** or **Enter** an * appears at their current position and their response is recorded. The response is calculated as a number which is centered on 0 and is + and - the maximum value specified, where the maximum value can be no higher than 39. !NUM,R,,,20 would make a number line 41 characters long (including the *), and store numbers from -20 to +20. (The plus sign is not put into the data.)

Text can be placed above the line if you like (e.g., Poor .... Excellent) by specifying it as part of the question text. The number line is presented in the middle of whatever box specification has been specified on the question, so you can move it around on the screen.

NUM,R cannot have anything but a maximum value specified. The maximum cannot be greater than 39. The line can therefore be a length from 3 to 79.

```
EX:
!NUM,R,,,30
```

2.4.6 Variable Length Question Type

Variable length, or VAR, questions are for controlled-length, open-ended responses up to 76 characters or as placeholders in the data (up to 240 characters wide). With a VAR question you can specify the minimum and maximum number of characters that will be allowed. The response will be stored in the data exactly as typed. However, the program checks that the response meets the criteria of the entry after stripping out any blanks before storing the response.

The interviewer will see underscores (____) on the screen for the minimum (or required) number of characters and periods (.....) up to the maximum number of characters.

The syntax for a VAR question type is:

```
!VAR,subtype,maximum number of characters,minimum
```

Specifying a subtype, or minimum or maximum number of characters is optional. Following the order of the above syntax, commas must be used as placeholders when an early item is not includedbut a later item is.

The default subtype is no subtype (i.e., allow any type of character).

The VAR question subtypes are:

A Used to create a marker for text that will be used in other questions. Like the CAT,A and FLD,A the VAR,A is hidden from the interviewer. A VAR,A gets its response from a previous GEN, SPC, or PHONE statement or by placing it on a previous question's data. The response from the VAR,A can then be displayed in the text of a subsequent question. Having a VAR,A with a minimum greater than zero will cause Survent to blow with error message #109 if the data is blank.

NOTE: \| performs a similar function. (See *2.5.3 DISPLAYING PRIOR RESPONSES AND DATA, Displaying Contents of A Data Location* for more information.)

B allows the question to be unanswered (and thus blank in the data). Use this with caution because it is difficult to track the interviewers' movements if some questions are blank.

D accepts date/time string and return "yyyymmddhhmmss"

You can enter a date/time as if you were at the "Enter time to call" prompt and Survent will return the julian date/time in the data. You can then use this date/time to control the !PHONE,S,104(or 160-179) statement's time to call. This means you can format your own screens to get the time to call as well as control the input to be within certain parameters, etc. The syntax is {label: .14 !var,d}. The width must be at least 14 to accommodate the returning Date/time:

Example:

```
{dattim: .14
      Enter the time to call the respondent in
      respondent's time
      !var,d}
      {!phone,s,104,dattim}
```

NOTE: to specify more characters as input, set the length longer. The datetime returned will be left-justified in the location field.

NOTE 2: You can only give specific date/times. That is, you cannot use "+# days" or "+# mins" for instance, or any of the "approximate time" keywords like "tomorrow" or "next week". You can, however, use "monday", "tuesday", etc and it will return the current time on that date.

Also, if you enter nothing, you get the current date/time returned.

This can also be used in conjunction with !PHONE,S,104.

E checks the input for a valid e-mail address. The requirement is that an "@" must be present followed by letters and at least one period (".").

L allows alphabetic character responses (A-Z), periods ("."), dashes ("-"), and blanks only. No numbers or other special characters are allowed.

N allows the entry of numeric responses only. This is useful with data such as telephone numbers or ZIP codes.

P allows the entry of phone numbers only. Any parentheses, dashes, spaces, or leading “1’s” will not be recorded in the data; the number of digits remaining must match the phone number size as specified in the phone file. If you back-reference the VAR,P question, you will see the number as originally entered.

As another option, you can allow an interviewer to type more characters than are stored in the data when using !VAR,P. This is because !VAR,P strips miscellaneous non-numeric characters from the input and only saves the number. The new syntax is:

Syntax:

```
!VAR,P,<Width saved>,<Minimum width saved>,<Max # characters typed>
```

For example, !VAR,P,10,10,20 would allow interviewers to type up to 20 characters, but it would require that the numeric characters be exactly 10. When they enter data, such as:

Example:

```
     (415) 777-0470
```

If you back-reference the question, you see exactly "(415) 777-0470". But, if you look at the data, you see "4157770470". You can later use this phone number to feed it into a !PHONE,5 statement and call the number.

Q does not echo the typed response (is quiet). This is useful for passwords. Asterisks are displayed on the screen instead of the actual typed response.

Specifying a minimum and/or maximum length is one way of ensuring that interviewers enter an appropriate response — the program will not accept a response that is shorter than the minimum or longer than the maximum. The default for minimum is zero and the default for maximum is one.

By specifying the same length for the minimum and maximum, you can ensure that an exact number of characters will be entered. This can be useful for same-length data, such as telephone numbers or ZIP codes.

VAR questions have no response list. The subtype and minimum and maximum lengths are the ways to limit the allowed response. Blanks are removed from the front of responses before they are stored.

Here is a sample VAR question:

```
EX:
{ZIP:
Please tell me your 5-digit zip code
!VAR,N,5,5 }
```

This is a subtype N, so only numeric responses will be allowed. The minimum and maximum lengths are both five characters, so exactly five numbers must be entered by the interviewer.

The EDIT statement allows the modification of a VAR or TEX response. (See *3.1.4 DATA ENTRY QUESTION FUNCTION MODIFYING STATEMENTS* for more details.)

Advanced Users Note: The 76-column limit is on data input for the VAR question. You can GEN,M to a VAR question with a width up to 3000 columns, or back-reference VAR,A questions with up to 3000 columns.

2.4.7 Text Question Type

Text, or TEX, questions are similar to VAR questions in that they are used for open-ended responses.

However, responses to TEX questions can be up to 5000 characters long. This will be further limited by the size of the box you describe, or the actual number of columns available in the text area of the data file.

TEX questions can accommodate very long responses *or* very short responses. They store the data in the most efficient manner, only storing what is typed in a compressed, variable format. Blanks are removed from the front of responses before it is stored.

The syntax for a TEX question type is:

```
!TEX,subtype,# of lines,# of columns (full-screen
mode)
!TEX,subtype,# of characters (line mode)
```

Specifying a subtype, or number of lines, columns, or characters is optional.

The TEX question subtypes are:

A is used to give a !TEX identity to a location containing text pointer data from a previous TEX question (see *GEN,M and SPC,P CODINGMODE*). Like the CAT,A, FLD,A and VAR,A, the TEX,A is "hidden" from the interviewer. This TEX,A response then can be back-referenced in the text of a subsequent question.

B allows you to use an **ESC** (DOS, UNIX in full screen mode) or **Enter** (in line mode) as the only response; by default, at least one character must be entered. **Enter** in full screen mode will not generate a blank response; it will just move you to the next line in the fullscreen edit box.

L forces line mode (UNIX). Line mode reduces the amount of CPU normally used in fullscreen mode; this is useful when user applications are running slowly or if you prefer not to use the full-screen editor. If you later EDIT this TEX question, it will be edited in line mode as well.

DOS: The TEX question employs a full-screen editor for the response. This allows the interviewer to

back up, re-enter text, use tabs and other editor functions while entering the response to the

question.

UNIX: The TEX question employs a full-screen editor for the response, like in DOS, but to move

around in the box you use **Ctrl-U**, **D**, **R** and **L** (for up, down, right, and left) instead of DOS's arrow keys. See subtype L to force line editing.

UNIX Line Mode: The TEX question employs a line editor for the interviewer response. The only editing allowed is backing up using the backspace key while still on the same line. See *4.2.2 RESPONDING TO SPECIAL QUESTION TYPES, Edit Mode* for more information on editing.

In full screen mode, the number of lines can be from one to the number of lines on the screen minus two. UNIX is limited to 21 lines, for example. The size of a line is determined by the screen control box size and the number of columns specified (see *2.5.1 SCREEN FORMAT CONTROLS, Using Boxes To Specify The Screen*

Position). The number of columns can be from 10 to 78. The number of lines and columns defines the box for the text response. The default for a TEX question response box is 9 lines by 78 columns (702 characters).

In line mode, you can specify the maximum number of characters allowed. (You also can use !TEXT,,5000 or !TEXT,,100,50 (to have 100-row by 50-column boxes. If you try to use this option in Terminal mode, however, Survent will blow.)

The data from TEX questions is stored at the end of the data file in a compressed format for efficient use of the data file's space. The TEX question itself holds a pointer to its response data at the end of the file, and must have a length of 1 for the pointer.

WARNING: If you put another question over the text pointer, that data may be lost. Use the CHK file to verify that you haven't done so. Overwritten TEX pointers can be recovered using the RAWCOPY program (back-pointers only) or with the example FIXTX^SPX spec file (all recoverable text problems).

A TEX response takes as much space as it needs in the data file on a case-by-case basis. This is in contrast to VAR questions that must reserve enough space in the data file to hold a response as long as the specified maximum length. If a TEX question is not asked, it will only use up one column, the column for the pointer.

Text data will start in the column indicated on TEXT_START on your header statement or, if you haven't used TEXT_START, in the first column of the next 80-column record after the last data column used. That column through the end of the data file is available for text data. You can fit approximately two times the number of typed characters as the number of data columns available.

PREPARE doesn't check to see if you're allowing yourself enough room for TEX responses. If you don't have at least 100 columns available, you will not be able to enter *any* TEX responses.

Data from TEX questions can be put in the regular data area using the GEN,M statement and a VAR question placeholder; for instance, !GEN,M,VAR1,TEX1 will move data from question TEX1 to the VAR question VAR1. Each VAR question can be 3000 columns wide, so you may need two VAR questions to hold the data from one TEX question (See *3.1.3 GENERATE SUBTYPES FOR GENERAL USE.*).

TEX question responses can be displayed with the LIST utility (see the *Utilities* manual), recoded with the RECODE utility (see *5.7 RECODE*), put out as ASCII characters in a file with the REFORMAT utility (see *5.8 REFORMAT*), and displayed or modified using CLEANIT (*Utilities*).

Here are two sample TEX questions:

```
                        EX:
                        {REASON:
                        What is your main reason for using the store
brand?

                        !TEX }
```

In this example, no subtype is specified. The interviewer could type from one to approximately 750 characters as a response to the question.

```
EX:
{REASON:
What is your main reason for using the store
brand?
!TEX,B }
```

This is the same question as above except it is a subtype B which allows no response — the interviewer could press **ESC** (DOS) or **Enter** (UNIX) if the respondent will not provide an answer, or if this question is being used for an interviewer comment and they have none.

TEX questions have no response list.

See *2.6.4 USING FUNCTIONS IN CONDITION STATEMENTS, Response Counting Functions* for the CHECKTEXT function that lets you determine the status of a TEX question: not asked, asked and answered, or asked and not answered.

See *4.2.1 RESPONDING TO DIFFERENT QUESTION TYPES* and *4.2.2 RESPONDING TO SPECIAL QUESTION TYPES* for more information on answering TEX questions.

2.5 QUESTION TEXT

The question text follows the condition statement, if one exists, and the question label line. The text will be presented to the interviewer as you format it on the line(s). By default, text beyond column 128 of the spec file will be truncated. You may allow more text on a line by using the SPEC_WIDTH parameter on the header statement; you may set this up as high as 300. The program will wrap text that is wider than the interviewer screen and less than the SPEC_WIDTH value in the header statement, unless you specify otherwise or are using box specifications. If the question needs more than one screen, the first screen will be presented along with a message to press **Enter** to continue.

Question text is optional; however, you will usually want some text to appear on the screen for data entry or display questions. Text is typed exactly as it will appear on the screen. If there is a prompt for data entry, by default it appears at the left edge and below the question text (and response list, if it exists).

You can include blank lines for readability. The text ends at the question type statement.

```
EX:
What is your marital status?
```

This would display the line “What is your marital status?” on the first line of the screen.

2.5.1 Screen Format Controls

Additionally, question text and text in the response list can have many special formatting commands in them. These are specified by typing a backslash (\) followed by the command syntax. The formatting options are explained below:

Start a New Line (\N)

You can start a new line on the screen by just starting your text on another line (for question text only) or you can type a backslash N (\N).

```
     EX:
Which do you like best, \Napples, \Noranges or \Nbananas
```

The program will skip to a new line on the screen wherever this is typed.

```
Which do you like best,
```

```
Which do you like best,
apples,
oranges or
bananas
-->
```

This would display apples, oranges, and bananas on separate lines. Using two sets of \N would leave a blank line in-between. To leave a blank line in the response list, use >N at the end of the line, or use the "=" equal sign response (see *2.4.2 CATEGORY QUESTION RESPONSE LIST, Equal Sign Response Code).* \N can also be handy in response lists to force a long response onto multiple lines, thereby allowing more columns of responses across the screen.

Clear the Screen (\T)

Use this as the first thing on the first line of question text to clear the screen prior to the question.

This is the default unless using box specifications. It can be used with a box specification to clear the screen prior to using the box (see *USING BOXES* below).

```
EX:
\T\(10)This cleared the screen before putting me
on line 10.
```

If specified in a different spot than described above, the clear screen will not work.

Append to the Bottom of the Current Screen (\A)

This allows you to put up questions in whatever screen space is available below the previous question. The screen is not cleared between questions. If the question does not fit in the space available, the screen will scroll up until it has enough space to accommodate the question. This will also reformat your response list to fit on the screen.

```
EX:
\AHere's another question to put under the
previous one
```

NOTE: Whether you put the \A on the first question text line by itself or immediately followed by question text, the text will still print on the first available line of the screen. Leave a blank line after the \A to force a skipped line on the screen.

Specify the Screen Position for text using Box Specifications \(#,#,#,#)

Boxes are used to specify where on the screen a specific question is to appear, or where to put the following lines of text. This can be used if you wish to put multiple questions on the screen, or to position the screen for this or all subsequent questions. No actual box lines appear on the screen unless you also display a border using \O# (see next section).

The syntax for a box specification is:

```
\(Top row, Leftmost column, Bottom row, Rightmost column)
```

You need only specify the parameter(s) you wish to change. The defaults are 1, 1, 24, and 80 in DOS, and 1, 1, 23, 80 in UNIX. You can specify a permanent box to stay in effect until another permanent box specification is seen. To set a permanent box you specify two left parentheses before the box specs. ***NOTE:*** In order to restore use of the full-screen, use \((1,1,24,80). A permanent box simply means you are changing the default box (or part of the screen) Survent can access. This box can be overridden on any question by supplying a different box specification. When no alternate box specification is given, it reverts to the permanent box.

Put box specs for the whole question at the beginning of the text. Put box specs for lines within the question at the beginning of the line. Text started on the same or next line will appear on the first line of the box on the screen.

```
                EX:
\(5)            Start question on 5th row; since no column is
                specified, it will start in column 1 and use the
                rest of the screen.

\(5,10)         Start question on 5th row; 10 is the leftmost
                column.
\((5,10)        Start this question and subsequent questions on row
                5, column 10.
\(5,,20,40)     Start question on 5th row; the leftmost column is
                the default (either 1 if no permanent box is set,
                or the leftmost column specified on your permanent
box spec), 20 is the bottom row and 40 is the
                rightmost column.
```

To specify the bottom row of the screen (platform independent), the syntax is:

```
\(BR)
```

The above syntax will print the text following it at the bottom of whatever screen your application is running on. Specifying \(BR-#) will print # lines above the bottom of that screen. This syntax can be used as part of the box specification.

```
EX:
\(BR-2,5,BR,40)
```

NOTE: This command does not keep the contents on the screen across questions. Instead, specify a permanent box for your other questions. Using boxes does not affect anything already on the screen outside the box. See \T above to clear the screen first. Also be aware that if an interviewer backs up to a screen with a box, the area outside the box is similarly unaffected.

If a question does not fit in a specified box on the screen in Survent, it goes into scroll mode starting back at the top of the screen. This is to make it easier to do interviews without worrying about what your screen size is set to. For permanent boxes, the screen control will continue on the next question.

Also see *2.4.2 CATEGORY QUESTION RESPONSE LIST, Positioning The Response List* for information on a screen positioning box for the response list for CAT and FLD questions.

Modifying Text in Questionnaire to Change Screen Display

You can now modify text in a file and it will change the text in the questionnaire the next time an interviewer sees that screen. The syntax is:

Syntax:

```
    \|&<FILENAME>|
```

Where "filename" is the name of a file. This means you can make text changes in your questionnaire without re-compiling it and putting up a new version. If you know in advance that you will have text changes somewhere, you can use \|&filename references and make changes on the fly.

Displaying a Border Around Question Text and/or Response List

The syntax for outlining the whole question is:

\O#

The syntax for outlining the recode table is:

\OR#

# options for both are:

1 - display a 1-line border

2 - display a 2-line border

3 - display a 3-line border (displays as a solid, thick line)

Terminals: You will only get a dashed single-line box regardless of the # you use.

You can combine these with a specified box for the question in this order: \O1\OR2\(5,5). This indicates the border for text, border for recode table, and box for question. This specification would outline the text and recode table separately. Optionally you could outline just the recode table:

```
EX:
\OR3
question text
```

```
\(5,5)
```

The following example outlines the entire question and defines a box for the question.

```
EX:
\O1\(5,5)
```

The border for a question outlines the entire size of the box, unless no box is specified; in that case, there is no border. If you want each question to be outlined, you will have to specify \0# at the beginning of each question's text.

NOTE:

- The \O or \OR must be specified at the beginning of the question text.
- If the question is not displayed in Highlightcats mode, you must insert a blank line after the end of the recode table or the Survent prompt will be inserted into the bottom line of the border. You could also position the prompt in another location using _ .

The \O option uses the screen area outside of the specified box to draw the outline. Thus, using \O with a box spec that does not specify all four points of the box, or which uses the outermost points of a box will cause a portion of the box to have no visible border. For example, using "\O\(5)" will display a horizontal line on the fourth line of the screen, and no other outlining.

Setting a default box of one less than the one used for non-outlined questions will allow you to specify outlined boxes using a box spec that does not include all four points. For example, setting the default box to "\((2,2,BR-1,79)" will make it so that a later spec of "\O\(5,10)" will have a complete outline of the question instead of just a single horizontal and single vertical line. Using "BR-1" instead of an exact number will help avoid problems related to differences in the length of the screen display across the various platforms on which Survent is supported.

Centering on the Screen (\OC)

To center the question text horizontally on the screen (or in the box), use \OCT at the very beginning of the text.

To center the response list on the screen (or in the box), use \OCR also at the very beginning of the question text.

Centering is especially nice when using Highlightcats mode — the cursor is placed on the response list in the middle of the screen.

The centering will be based on the current box in effect.

Specifying Default Boxes and Screen Centering Options

You can also specify default boxes and screen centering for the following questions. In this way you won't have to re-specify it from question to question. The syntax is "\OD<options>" and the options are:

Option	Result
\OD1	Puts a thin box around question, with separate boxes around text and response list; 2 and 3 for medium and thick boxes.
\Odct	Centers the response text (not response list)
\Odcr	Center response list (not the text)
\ODt1	Puts a thin box around text of question, use t2 and t3 for medium/thick boxes
\ODr1	Puts a thin box around response text, use r2 and r3 for medium/thick boxes
\OD0	Turn off all boxes and centering

You may combine the text, centering text, and centering response list options. This is most useful in DOS applications where boxes are often used on the screen. In UNIX systems, there is only one box display type, so the t2-t3 and r2-r3 options will not give you a thicker box.

Positioning the Data Entry Prompt (_)

The prompt is the dashes and arrow (-->) where you enter a response on data entry questions (except those in Highlightcats mode). The default prompt position is one line below the bottom left corner of the text (or response list). If you specify "_" in the text, the cursor will be placed there, in place of the _, and the standard prompt will not be displayed.

```
EX:
How many apples are you growing this year? \_
```

This would put the cursor to the right of the text. You may follow the _ with characters and enclose the whole thing with a text enhancement to make it stand out.

```
EX: {
What is your zip code?\U\_____________\E
!VAR,N,5,5}
```

This would display the area where the response will be entered as underlined.

```
What is your zip code? _____________
```

NOTE:

- An enhanced response area (e.g., bold or flashing) causes the response to also be enhanced.
- VAR questions using the \- prompt will no longer display the underscores and dots to indicate response length.
- Using _ for TEX question will position the box (in screen edit mode).

Horizontal Response List Wrapping (\H)

By default, response lists wrap vertically. Using \H at the end of the question text allows you to present the response codes in horizontal order on the screen. This is used on CAT and FLD questions only.

```
EX:
\H
! CAT
1 EXCELLENT
2 GOOD
3 FAIR
1 EXCELLENT 2 GOOD 3 FAIR
```

This would display the three responses across the screen, rather than down. You could combine this with Highlightcats mode to get a highlighted horizontal response list.

2.5.2 Text Enhancements and Graphic Characters

Text enhancements are used to highlight or emphasize specific text. An enhancement is turned on with a backslash-letter code, and turned off with a backslash-E. You can set Enhancements using a mouse in the Script Composer. Using specifications, the to-be-enhanced text is typed between the two back-slash codes. The codes and their enhancements are:

\B	**Bold**
\F	Flashing
\I	Inverse video
\C	Color
\U	Underscore
\E	Turn off enhancement

Activating an enhancement will affect all the text that follows until it is turned off with \E. This will stay in effect across lines of question text or on multiple lines of a wrapped response list item. In this example, the word “not” will be bold:

```
EX:
How many children are \Bnot\E living at home?
```

How many children are **not** living at home?

You can combine enhancements where they are compatible. For instance, you can use both bold and underscore: (note that the \E turns off all enhancements active at this point)

```
EX:
How many children are \B\Unot\E living at home?
```

How many children are **<u>not </u>**living at home?

NOTE:

- If both inverse and underscore are chosen, you will only get underscore. If you forget to turn off an enhancement, it will carry over to subsequent text lines.
- On terminals, you will only get underscore, flashing, and inverse video. On both monitors and terminals, you will be further limited by what that machine supports.

When running on a color machine, flashing and bold enhancements work in the same way. Inverse will display in either the inverse of your system colors, the CfMC screen color set on the >COLOR command (i.e., the first color spec specified on this command) or on the COLORS environment variable. Underline will display as white on green, but underline and inverse combined will display as white on cyan. Underline and flashing combined will display as flashing white on green.

When running *color* specs on a non-color machine, flashing and bold work in the same way, and all color specifications are displayed as inverse.

Color Enhancements

Color enhancements can be set with \Cfb, where f is the foreground and b the background color. If your UNIX terminals or DOS PCs are properly equipped, you will be able to set the foreground and background colors. In UNIX, you must specify "setenv COLORS on" in the interviewer login script for the colors to be displayed to them. In DOS, you just need a color monitor.

You may also specify a color specification for the entire question or for the entire interview, using the \D character. The syntax is:

```
\Cfb          Color foreground/background
\DCfbCfb      Default color for question; 1st Cfb
              set is for text; 2nd Cfb set is for
              the responses
\DDCfbCfb     Default color for this and subsequent
              questions
```

The foreground and background colors can be any of the following colors:

Z	Black
W	White
R	Red
G	Green
B	Blue
Y	Yellow/Brown
C	Cyan
M	Magenta

NOTE: You cannot specify the same color for foreground and background because the text and background will blend and be unreadable.

In the example above, the question text will be black on white background and the responses will be blue on brown background.

```
EX:
How many children are \CYRnot living at home\E?
```

In this example, the words “not living at home” will be brown on a red background. You can also use the text enhancements above (Bold, Flashing, etc.) in combination with colors. Bold applied to a color often makes it appear to be a different color, rather than just bolder (e.g., brown letters become yellow).

```
EX:
How many children are \B\CYRnot\E living at home?
```

In this example, the word “not” is in bold and will be yellow on a red background.

Bold can also be specified by using a plus (+) after the C and before the color spec.

```
EX:
\DC+RWC+WR
```

This will give you bold red on white text and bold white on red responses.

PREPARE will strip color specs in the HRD file and replace them with the **ESC** sequence for bold if a printer type is specified.

Use an underscore after the color specification to separate the command from an adjoining word starting with a “C” which the program might try to read as an additional color specification (i.e.,\CRG_choose).

Showing Text Only in webSurvent/webCATI

You can have text that is specifically targeted for webSurvent and webCATI. only. The syntax is:

```
Syntax:
\+? (this is webCATI-only text)
\+! (this is webSurvent-only text)
```

This is an extension of the “\+ Survent display only” feature.

See *Alternate text* in *section 2.5.4* for more information.

Displaying Graphics or Other Special Characters (\^)

Graphic characters, control sequences and escape sequences may be included in the question text by specifying \^ and the one or two required characters.

Each ASCII character has a corresponding hexadecimal code. For non-keyboard characters, you *must* use hexadecimal codes to see the corresponding characters. For some characters such as \ or } which are read as commands to the programs, the only way you can print them is by using the hexadecimal code. To do so, use the slash and caret (\^) and the two-digit hexadecimal value corresponding to the desired symbol (see *Appendix E: SPECIAL CHARACTERS* for a chart displaying the values and symbols).

```
EX:
That concludes our interview. Thank You \^01
```

In this example the ^01 is viewed as a smiling face that appears in the text after 'Thank You' (monitors only).

Control sequences require at least one space following the character code.

```
EX:
Good work, Interviewer! \^G
```

This example rings a bell (**Ctrl-G;** on terminals only) after the text is displayed.

Escape sequences require a left bracket before the characters.

```
EX:
That concludes our interview. \^[QS Thank You
```

Refer to your operating system or terminal manual for further information on control and escape sequences.

2.5.3 Displaying Prior Responses and Data

Displaying prior responses in the text allows you to personalize questions without expecting interviewers to remember previous responses. This is referred to as back-referencing the response to a question.

Displaying Responses from Single Response Questions (\:)

To back-reference a single response question, use this syntax:

```
\:label or \:QQ#
```

Back-referencing shows the answer for the question when it was executed. If the question has not been asked, nothing will be displayed, even if there is data in the same location as the question.

If the question was asked, back-referencing the question will display its response, unless a question with ALIAS=<same label> has been asked between the original question and the question you are back-referencing, in which case its response is picked up (See *2.3.5 QUESTION LABEL LINE, Question Options*).

If the question was asked and there was no alias question, the data and the back-reference will be the same for NUM, EXP and VAR questions. To avoid confusion, you may want to use \|qname for VAR, NUM and EXP questions.

If you are back-referencing a CAT or FLD question, the first response's text is shown (not the response code). For TEX questions, the text of the response is shown.

NOTE: If the response list is made up of codes only and no text, then you must display the contents of the data location (\|) since the back-reference will otherwise be blank.

Here is a sample question that back-references a previous response:

```
EX:
Of these \:NUMCHILD children, how many are in
college?
```

The response from question NUMCHILD would be inserted in the text of this question. Note that the label must start with a colon (:label); an ending colon after the label is optional.

The space used on the screen by the response is variable depending on the actual length of the response. You can control the amount of space used by appending a caret (^) and the number of spaces to use.

```
EX:
Brand A - \:BRANDA^10 is the favorite.
Brand B - \:BRANDB^10 is the second favorite.
```

In this example, the display of the text from question BRANDA and BRANDB will use 10 spaces on the screen. By specifying a width you can keep the text to the right of BRANDA and BRANDB's response lined up properly. If the text is longer than the width specified, it will be truncated. You can use a dash (-) before the number of spaces to right-justify the text.

```
EX:
Brand C - \:BRANDC^-10 is the next favorite
```

NOTE: Leaving spaces between the label and the caret will not affect screen presentation but will allow you to line things up as you're writing them.

```
EX:
NAME: \NAME ^-20
ADDRESS: \:ADDRESS1 ^-20 (home address)
\:ADDRESS2 ^-20
\:CITYSTAT ^-20
```

You can also reference a previous response by question number, using a QQ# reference (e.g., \:QQ#).

```
EX:
Here is \:QQ20
```

For more information on back-referencing, see *3.1.4 DATA ENTRY QUESTION FUNCTION MODIFYING STATEMENTS, LOOP/END_LOOP Statements*.

NOTE: Do not back-reference soft-coded question numbers (i.e., question numbers you have not coded your self, but picked up from a CHK or SUM file.)

Displaying Responses from Multiple Response Questions (\#)

You can reference any or all of a multiple-response CAT or FLD question’s responses. The response text for each response chosen is displayed on the screen, separated by commas. The syntax for referencing a multiple-response question’s response is:

```
\#(label,first response,number of responses)
```

Only the question label is required. To see all the responses to a question in the order they were entered, use \#(label).

```
EX:
The cars you own are: \#(CARS)
```

This would show all the responses to the question CARS. To see only specific responses, enter the first response and number of responses parameters.

```
EX:
The two best cars you own are: \#(CARLIKES,1,2)
```

This would display the first two responses to the question CARLIKES, separated by a comma.

You will not get a spec error if you incorrectly specify more responses than available.

```
EX:
\#(name,4,3)
```

The above example will show up to three responses starting at the fourth response.

If you are using an alias question to join an “Other (specify)” response to the pre-listed responses, the “Other (specify)” response will be treated the same as the pre-listed responses. What is being displayed is coming from the answer array, not the data.

The **answer array** is a memory area where Survent keeps track of answers given previously so you can do back-references to them. When you compile your specs, PREPARE makes a note of the questions that are used in back-references in \:label references, SPC, 9 question types, as loop controllers, or TEX questions that are EDITed so that Survent can store their answers as they are entered. For related information, see the header statement option ANSWER_LENGTH.

Normally only the first response to a question is stored. If the question is used as a loop controller or is referenced with the \#(label) syntax, then all the responses to that question are stored in the order of mention. If you are using alias names, then the last alias question will overwrite any previous answer, unless you are storing all the responses (as above), then each alias response is appended to the end of whatever responses already exist. Use (O=label) to get the other response inserted in proper order in the answer array.

If you wish to save the order of mention on a multiple-response CAT question you can use a LOOP to save it into the data. Blanking out the data does not affect the answer array, nor does backing up in the questionnaire using a backward GOTO or a backward SKIPTO. A GEN,E can erase the answer array of a particular question should you need to.

Displaying Contents of a Data Location (\|)

You can display data directly with the \| syntax. This is useful if you have a counter and you want to know the current value without putting a VAR,A on top of the location first, or when referencing existing data in Codingmode, or using PHONE,G statements to get sample data.

The syntax for displaying data directly is:

```
\|location|
```

The backslash is followed by a vertical bar (|), a location specification and another vertical bar. The ending vertical bar is optional. The location specification can be an absolute location or record/column location and width (i.e., \|col.wid|), or a location specified as starting column - ending column (1/25 - 1/30). Where no length is specified, it defaults to 1. Or, it can be a prior question label surrounded by colons \|:label:|.

```
EX:
You said you had \|3/20.5| Bentleys, is that
correct?
You said you had \|3/20-3/24| Bentleys, is that
correct?
```

Both of the above examples will echo what is in the data in record 3, columns 20-24.

```
     EX:
You said you had \|Numcars| Bentleys, is that correct?
You said you had \|Numcars Bentleys, is that correct?
```

This will do the same thing as the prior example, if Numcars was in location 3/20.5.

You may also specify a ^# or ^-# after the location. This will control the number of spaces allotted to the answer on the screen. By specifying a width, you can keep text to the right of the answer lined up properly. If the data is longer than the width specified, it will be truncated. Using -# will right-justify the answer within that field width.

```
EX:
You have \|1/10.5|^7 cars, is that correct?
```

NOTE: Leaving spaces between the ending vertical bar and the caret will not affect screen presentation but will allow you to line things up as you're writing them.

Displaying Data from the Phone File and Suspend Text (\[ )

"\[" can be inserted in a text line to recall data directly from the phone file. See PHONE,G in *6.3.1 THE PHONE STATEMENT, PHONE, letter Subtypes* for the phone file layout to use. Any valid locations or phone variable names can be specified. For instance, you may use \[23.2] or \[time_zone] to get the time zone the phone record is in displayed.

To display suspend text, we use a location beyond the maximum location in the phone file. Use "\[3001.80]" to display the suspend text in version or earlier (old small phone file format). In version 7.6l or later, use "\[20001.80]".

Displaying Environment Variables (\!)

You can read environment variables into your questionnaire by using \!. You make environment variables by saying (at the operating system prompt):

```
setenv name value (UNIX)
SET name=value (DOS)
```

By specifying \!variable name, the current setting of the variable will display on the screen or be put into the data if using an SPC,9.

2.5.4 Miscellaneous

Displaying Question Label (\@)

\@ is used to display the question number or label in the text of the current question.

If the question has a label, the label is displayed (whether or not there is a question number). If there is no label, the question number will print (whether or not there is a user-assigned question number).

Alternate Text (, \+, \-)*

You can store different text in the DB file and DEF file from the text that displays on the interviewer's screen by using the alternate text feature. Text stored in the DB file is used for later reference by the utilities or Mentor when printing reports or making tables, and the DEF file is created by ~PREPARE or REFORMAT when making variable labels for Mentor or other data processing packages.

\+ Text following a \+ will print on the Survent screen but will not be stored in the DB or DEF files. This affects all text after the \+ until the next * or \- (or the end of that text item).

\- No text following a \- will appear on the Survent screen until the next * or \+ (or the end of that text item). This text will only go to the DB or DEF files.

***** All text following a * will go to both the Survent screen and the DB and DEF files. This is the default, and is only needed to turn off a preceding \+ or \-.

Displaying (\) backslashes

By default, the backslash character is considered a text modifier and it is used as such with backreferencing text.

However, there are times when you want the backslash to be displayed as a backslash and not to be used as a modifier. For example, if you are asking for a DOS filename to be entered, the respondent might enter:

```
Beststf\data\easy.tr
```

By default, Survent would treat the \b as bold, \e as end, etc. If you want Survent to backreference a response and show the backslashes instead of interpreting them, use the “=” character before the reference. Here are some examples:

```
\=:filename (show the filename)
\=#q1 (show the responses to q1)
\=!datetime! (show the system variable datetime)
```

2.5.5 Specifying Multiple Languages (\L and [\L=])

Questions and/or response text may be displayed in one of many languages. In the questionnaire, you specify "\L<language code> text" to start a section for a particular language. You may switch back and forth between languages at any time on a question. Interviewers may switch between languages at any time by using L=<Language code>, or you may use the {!SYS,L,<language code} statement in the questionnaire to switch the language.

If you do not specify a particular language at the start of your text or on a response code, the program displays the text for all languages. You can also display all unmarked text in all languages by using "\L**".

If you use the compiler directive DEFAULT_LANGUAGE it defaults to the first language on the "set=()" list, but you can change it at any time. This would be the language used if you didn't specify \L on some text. If you set the default_language to "!!" you get all text including the "\Lxx" part of the text (for debugging problems).

Survent supports many languages. It can use up to 500 different languages. In order to do so, two-character codes that represent each language must be used. Most questionnaires will only use a few languages. For this reason, Survent allows you to convert from two-character mode to one-character mode for ease of programming.

To use single-character mode, you would enter your !LANGUAGE statement as follows:

```
EX:
{!LANGUAGE set=(E=EN F=FR S=SP) }
```

The first letter is the letter you will use when specifying the language. And, the second is the actual CfMC code for that language. In this case, when you write \LS TEXT, you will be placed in Spanish mode.

If you want to program your questionnaire with two-character mode, your !LANGUAGE statement would look like:

```
EX:
{!LANGUAGE set=(E=EN F=FR S=SP) }
```

Therefore, you would specify "\LSP" to switch to Spanish mode. (This is assuming you are using one-character mode.)

When writing the questionnaire, you may switch languages at any time. The TEXT part of the question starts in "All language" mode. Once you switch languages, the language stays in effect for the rest of the text until another \L code is recognized. You may switch back to "All language" mode by specifying \L<blank>.

In the RESPONSE CODE areas, each new response code starts in “All language” mode until you specify a \L command or have a [\L=] response block. The same rules apply for \L in the response code area corresponding to the text part of the question.

```
EX:
INTERVIEWER, PLEASE ASK:
\LENWhat is your name? \LSPComo se llama?
\LDKWie heissen Sie?
```

In this case, interviewers will always see:

```
INTERVIEWER: PLEASE ASK.
```

Then, depending on the language mode, they will see one of the three languages in which they should ask the question.

Compiler directive CHECK_FOR_MISSING_LANGUAGES makes sure all languages are specified on each question that has any language specified. The default is not to check.

"\L**" accounts for all languages, unless there are other \L's in the text, in which case there must be one for each language in the set=.

Here's an example:

```
[study language=(set=(en,fr) check_for_missing_languages)]

      {Q1:
      \L**This is fine.
      !disp}

      {Q2:
      \L**And so is this.\LenEnglish\LfrFrench.
      !disp}

      {Q3:
      \L**But not this.\LenEnglish.
      !disp}
```

NOTE: All languages in the !language set must also be in the language set in the header.

Specifying Languages in Code Lists

There are two ways to specify languages on code lists. For example, with question text, you may use "\Lxx" to change to language xx on any particular code in the list. However, if you have many languages or many codes, or you wish to send the entire list to a translator, it is easier to keep all the codes together for each language. Here is an example of both:

```
      EX:
!FLD
1 \len English text\lfr French text\lsp Spanish text
2 \len English text 2\lfr French text 2\lsp Spanish text 2

      EX:
!FLD
[\len]
1 English text
2 English text 2
[\lfr]
1 French text
2 French text 2
[\lsp]
1 Spanish text
2 Spanish text 2
```

Although the first way contains less text, it is more difficult to deal with when translating because you have to copy and paste each line to its proper position on the response list. Using the new way, you can send the response lists out to separate translators and just insert the response list for each language (or use &file to read it) into the appropriate place on the list.

Changing the Language

As noted above, there are two ways to change the language being used:

Interactively, interviewers may type "L=<language code>". When this is typed, the current question redisplays in the language specified.

Programmatically, you may use the !SYS,L,<language code> command. By using this, you may ask the interviewer which language they want to use, and then execute a !SYS,L command based on that question.

The language stays in effect for later questions until changed again or until the end of the current interview. You may record the language currently in use by specifying the !SPC,7 statement, position 110.2.:

```
     EX:
{Currlang: !SPC,7,110,2 }
>PURGE_SAME
~PREPARE COMPILE
[LANG]
{!LANGUAGE SET=(E=EN F=FR S=SP)}
''E - English F - French S - Spanish
{LANG1:
Which language would you like?

!FLD
1 English
2 Spanish
3 French
}
{!IF LANG1(1)
!SYS,L,E }
{!IF LANG1(2)
!SYS,L,S }
{!IF LANG1(3)
!SYS,L,F }
{BEEF:
\LE Where's the beef? \LS \^a8Dond\^82 est\^a0 la
bisteca? \LF Ouest la bouef?
!VAR,,70 }
{MORE:
Everyone sees this \LF francais \LS espanol \LE english
\L* all languages
!DISP,1}
~END
```

If the language picked in the question LANG1 is Spanish, for example, at MORE you will see:

```
Everyone sees this espanol all languages
```

Invoking Specific CfMC Language-Error Messages

NOTE: The character(s) specified on the !LANGUAGE statement option must match those used on the L= command, \L and !SYS,L commands. Otherwise, the command will have no effect.

You may use a language code that is not supported by CfMC, but you will not get the CfMC-specific program error messages and prompts in that language. You will get the language text you specify in your questionnaire, but CfMC error messages will be in English in this case.

The CfMC language codes that are currently supported are:

DE: German

EN: English

JP: Japanese

FR: French

IT: Italian

PT: Portuguese

SP: Spanish

SV: Swedish

ZH: Chinese

You can add messages to the CfMC message file for your particular language for various program prompts and error messages, etc. using the MAKEMSG program. You may call CfMC Support for details on how to do this.

Using Multi-byte Languages

Some languages that are used require special handling by CfMC programs. These are languages that use characters that are either in the extended ASCII character set (Russian) or that use pictograms (Japanese or Chinese character sets).

In order to use these languages, you must specify the character set type as follows:

For Japanese Characters:

```
     EX:
{!LANGUAGE SET=(JP) CHARACTER_SET=Shift_Jis}
```

For Chinese Characters:

```
EX:
{!LANGUAGE SET=(ZH) CHARACTER_SET=Multi_byte}
```

Once specified, you may have these types of characters in the text of your questionnaire. Note that not all editors support these characters, and transferring files with these characters must always be done in binary mode. Also, to display these characters on a computer may require a special encoder.

WebSurvent, using Internet Explorer, can display these characters. However your PC or terminal will not be able to display them without special software. Contact CfMC Support for more information.

Modifying the text of CfMC messages

You may specifically control the text of CfMC messages for your language using the {!ERROR_MSG ####} compiler directive. This is generally placed at the top of the questionnaire. For example:

```
EX:
{!ERROR_MSG 1512 You MUST enter some text! }
```

This would display “You MUST enter some text!” instead of the default CfMC message whenever message number 1512 is invoked.

To control multiple language error messages, use the following syntax:

```
EX:
{!ERROR_MSG 1512
fr1512 Vous devez entrer un texte
1512 You MUST enter some text! }
```

This will display "Vous devez entrer un texte" instead of the default "You must enter text." when that CfMC error message displays in "French" mode and “You MUST enter some text!” when in English or standard mode. See the !ERROR_MSG statement for more information.

2.6 CONDITION STATEMENT

You may wish a question to be asked (or a statement executed) only if some condition is true. The condition statement lets you specify exactly what must be true in order for Survent to do the operation. You may use a previous question's response(s) or data to control the execution of any type of question or control statement.

Condition statements are also called IF statements ("If this is true, then do this operation."). Condition statements follow the question label line, and start with !IF followed by one or more references connected by logical operators AND, OR or XOR. Labels, question numbers and data locations may be used as the references. Parentheses are used to separate parts of complex logic conditions or calculations.

You can have up to 32,000 characters in an IF statement. Use an ampersand (&) at the end of the line to continue an IF statement to another line.

NUMITEMS and other functions referred to below are described in *2.6.4 USING FUNCTIONS IN CONDITION STATEMENTS*.

NOTE: Comparisons using =, < >, >=, and <= operators require that both sides of the equation match for type, and they must be either numeric or string variables. You can force this to work by putting brackets ([ ]) around a VAR type variable, for example, which converts the type to numeric or putting brackets and a dollar sign ([numtype $]) around a numeric type variable for a string comparison.

2.6.1 Referencing Question Types

The syntax for IF statement references varies depending on the type of question you are referencing.

CAT and FLD Question Types

The syntax for category (CAT) or field (FLD) question response list references is:

```
!IF label(response code 1,response code 2,etc.)
```

```
or

     !IF label(<>response code 1,response code 2,etc.)
```

Following !IF is a space, then the label of the question that the condition is based on. Enclosed in parentheses, and separated by commas, a period, or a dash, are the specific responses that trigger the condition. Commas (or nothing) separate specific codes (1,2,3), while a period or a dash indicate a range (all codes, from response code 1 to n; i.e., 1-5). You can use any of *code, code* or *code.code* or *code-code* to specify the responses. You can combine separate codes and ranges (1-5,8). "< >" at the beginning of the response list means *not* these responses. You may use the "^" placeholder in codes that are different lengths, eg. (Yes,No^).

```
     EX:
!IF BRAND(1,2,3) or !IF BRAND(1-3) or !IF BRAND(123)
```

For this example, a response to the CAT or FLD question BRAND of 1, 2, or 3 would trigger the condition and cause the question or statement to be executed.

```
     EX:
     !IF BRAND(<>1-3)
```

The example above shows the use of the not equal to (< >) characters in a response list. It says, If CAT or FLD question BRAND is not a 1 through 3, execute the question.

NOTE: When using the period or dash to indicate a range, be aware that 1.4 or 1-4 only means 1,2,3,4 if those response codes were in that order in the original response list. If the response list was in the order 1,5,4,3,2 then 1.4 means 1,5,4.

Use the XF(NUMBER_OF_RESPONSES()) function to count the responses to a multiple response CAT question.

```
     EX:
     !IF XF(NUMBER_OF_RESPONSES(BRAND)) >= 2
```

In the above example the question will be asked if the number of responses to the question BRAND is greater than or equal to 2.

NUM and EXP Question Types

For numeric (NUM) questions or expression (EXP) statements or other numeric data, the syntax is:

```
!IF label or constant operator label or constant
```

For numeric data, you may use labels or constants (or data locations; see *2.6.2 USING DATA LOCATION REFERENCES*). All labels *must* contain valid numeric data or the condition will be considered false. Following the label is a relational operator and then a number or another label.

The operators are:

=	equal to
<>	<> not equal to
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to

```
EX:
!IF AGE >= 21
```

A response to question AGE that is greater than or equal to 21 would cause the question or statement to be executed. To treat blank or non-numeric responses as if they were zero, use the X function. This is important because it allows the condition to be checked if one of the elements is missing.

```
EX:
!IF X(FEB)< X(JAN)
```

In this example, the question would execute if the response to FEB was less than the response to JAN. Without the X function, if one of the questions did not have a numeric response, the condition would not be considered true.

See *2.6.2 USING DATA LOCATIONS, ASCII Data Comparisons*, on how to refer to the NUM exception codes.

You may also look for numeric data in groups or ranges. The syntax is:

```
[label # num1-num2,num3]

EX:
!IF [Q3# 1-100,150]
```

VAR Question Types

For variable length (VAR), single response field (FLD) questions, or NUM with string data, the syntax is:

```
!IF label$ or "text" operator label$ or "text"
```

Note that the label is followed by a dollar sign ($). This tells the program that it is doing a string (or literal) comparison. The operator can be any of those listed under *NUM AND EXP QUESTION TYPES.*

The text or label to compare to follows. Text must be enclosed in quotes. If the specified text or the response to the second label$ matches the response, the condition will be true.

```
EX:
!IF OTHER$ <> " "
```

In this example, the question would execute if the question OTHER was not blank.

```
EX:
!IF LIKEIT$ = "yes"
```

This would execute the question if the answer to question LIKEIT was equal to yes. The comparison is not case sensitive — uppercase and lowercase text match — and leading blanks in string comparisons are ignored.

The program only checks as many characters as you specify. For example, if you specify “BUD”, the program will check only the first three letters. Consequently, any word or phrase starting with BUD would match, such as: BUDDHA or BUDGET. You could specify “BUD” to not match the other words (note the space after the ‘D’).

The <,<=, >, or >= operators use the ASCII code sequence to determine the relationship. All letters are upshifted before the comparison is made.

TEX Question Type

Text (TEX) questions, with text type data, cannot be referenced directly. This is because the responses to TEX questions are extremely variable. Instead, you can use the CHECKTEXT function to count the number of characters responded or whether the question was asked.

```
EX:
!IF CHECKTEXT(OTHER) > 1
```

If the number of characters entered for TEX question OTHER was greater than one, the current question will be executed. See *2.6.4 USING FUNCTIONS IN CONDITION STATEMENTS* for more information on the CHECKTEXT function. If you want to check the characters in the TEX question, use a GEN,M to move the TEX data to a VAR question, then use that for the conditional checking.

2.6.2 Using Data Location References

Specific data location references are useful when:

- You don't have a label at the specified location, i.e., it is within a question, or over multiple questions.
- You wish to use a data type not available for label references, e.g., ^^ or #.

- You wish to override the checking for the type f question being referenced. You could reference a response to a CAT question as a number, for instance.

The data location reference syntax is:

```
[location or label datatype datacodes]
```

There must be a bracket starting and ending the data location reference.

A location specification is required. To specify location, you can use a question label, question number or a specific column number. If using a label or question number, it must be for a question prior to the current question. If using a location, it may be in either absolute number format (103), or card/column format (2/23). In addition, a width may be specified, preceded by a period. The default width is 1 for data locations, or the width of the original question if using a label or question number.

You can also specify starting column - ending column (2/23 - 2/27).

```
EX:
!IF [10.2] > 9
```

If the number in columns 10-11 is greater than 9, t he statement would be executed.

The data type is not required. It defaults to numeric data type. [5.2] and [QQ23] would look in the specified location and expect a valid number.

The possible specified data types are string ($), punch (^ or ^^) and numeric range (#). Punch and range data types require data codes.

When using a label or question number for the data location of a question, you can use a column offset. The syntax for using a column offset is:

```
[label +/- offset.width^punches]

EX:
!IF [USE+2^5.0]
```

This example says if the data location two columns after the first column of question, USE had punches 5,6,7,8,9,0 then ask the question. (See 2.6.2 USING DATA LOCATION REFERENCES, Punch References for more information on punches.)

Punch References

The syntax for a data location punch reference is:

```
[location^punch code list]
```

Punch references are generally used when looking at CAT question responses, or specifically created punches created with generate (GEN) statements. To make punch references, refer to the label or column in question. Use the data type format (^) and then specify the code list. The specific binary punches referenced are 1-9, 0, X, and Y. Using the double caret (^^) allows you to specify punches across columns. In this mode, the first 12 punches are referenced as codes 1-12 (or as 1-9, 0, X and Y). The first punch of the second column is punch 13 and all second column codes are 13-24. You can reference up to 10 columns (120 codes) in one punch reference using the double caret.

Here is a display of the punches across two columns and their relative code number:

```
Column 1st 2nd Punch 1 2 3 4 5 6 7 8 9 0 X Y 1 2 3 4 5 6 7 8 9 0 X Y
Reference code 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
```

```
EX:
!IF [10.2^^15]
```

This references the 3 punch of column 11, the fifteenth possible punch in the two-column field beginning at column 10.

```
EX:
!IF [CARTYPE^10]
```

This references the 1 and 0 punches of the first column of question CARTYPE. The punch code list can refer to as many different punches as the width of the location specified allows. You can use commas between separate codes, or use a period for a range of punches.

```
EX:
!IF [10.2^^1,5,20,22]
```

This references column 10 punches 1 and 5 and column 11 punches 8 and 0. If any of these punches are in the data, the question will execute. You can also combine separate punch references and a range of punches.

```
EX:
!IF [10.2^^1.5,8]
```

If punches 1 through 5 or 8 are in the data in column 10, the question will execute. Additionally, the special characters N and B can be used. The N means *not* the following set of punches.

```
EX:
!IF [10.2^^N1.5,10-23]
```

If column 10 does not have a 1 through 5, 0, X, or Y punch, and column 11 does not have a 1 through X punch, the question will execute. The B is used to specify a blank set of columns.

```
EX:
!IF [10-11^^B]
```

This would cause the question to execute if columns 10 and 11 were blank. You can also combine the blank reference (B) with the not reference (N).

```
EX:
!IF [10.2^^NB]
```

This checks that columns 10 and 11 are not blank in order to execute the question.

NOTE: When using a question label, make sure you use the correct indicator (i.e., ^ if one column wide and ^^ if more than one column wide). If you only wan t to check for punches in the first column, specify "label.1^punches".

ASCII (String) Data Comparisons

String comparisons require a dollar sign ($) on the location reference, and use the string operators = (equal) and < > (not equal). Note that question labels can also use string comparisons without the required brackets, but in that case the question type must be VAR or FLD.

```
     EX:
     !IF [33.5$] = " "
```

or

```
     !IF [OTHER.5$] =" "
```

or

```
     !IF [33-37$] =" "
```

The first and third examples say, IF the data in the location columns 33-37 is blank, execute the question. The second example says, IF the data location referred to by the first five columns of the question OTHER is blank, execute the question. String comparisons are used for the exception codes in a NUM question.

```
     EX:
     !IF [FAMILY$]="DK"
```

ASCII (String) Lists or Range References

You can also look for string data in groups or ranges. The syntax is:

```
     EX:
     !IF [label$"string1"-"string2","string3"]
```

The number of characters in each of the strings must be less than or equal to the width of the question. The dollar sign ($) indicates you’re looking for string data (literals). The strings must be enclosed in quotes. If you use a range, put a dash (-) between the quoted strings, with spaces allowed. If using a range, the strings must be the same length. A comma separates specific items. An ampersand (&) may be used to continue the specification to a new line; as always, break at a logical point, use the ampersand, and continue on the next line, indenting for readability if you like.

Letters in strings are upshifted before the comparison is done (as is the case elsewhere in Survent). You can use non-alphanumeric references in the range; Survent uses the standard ASCII code sequence to determine what will fall in the range.

```
EX:
!IF [FIRST$"a"-"z"," ","."]
```

This example is seeing if the question labeled FIRST has characters from A to Z, a space or a period. The maximum length of any one string referenced this way is 30 characters. *Also see 2.6.1 REFERENCING QUESTION TYPES*, VAR Question Types for other examples of string comparisons.

Numeric Data References

Numeric data references follow the same rules as for the NUM and EXP question types regardless of the question type. If there is not valid numeric data, the result will always be false. Use the X function (see *2.6.4 USING FUNCTIONS IN CONDITION STATEMENTS*) to provide a valid 0 for nonnumeric data.

```
EX:
!IF X([20]) < 25
```

This dictates that if the data in location 20 for a length of 2 is less than 25, execute the question. Numeric data types use the numeric operators (>, <, =, < >, >=, <=) for comparison, and arithmetic operators (+, -, *, /) for math. Up to 4 digits of significance are read and used.

Number Lists or Range References

The syntax for a data location numeric range reference is:

```
[location #number range1, number range2, number range n]
```

The location is any valid location specification (number or label). Then comes a hash mark (#), followed by a range. The two numbers in the range are separated by a dash (-)/

```
EX:
!IF [40.3#30-50]
```

or

```
!IF [AGE# 18-64]
```

This statement says if the data in location 40 for a width of 3 has a number from 30 to 50, execute the question. The range can also consist of a combination of individual numbers and ranges.

```
EX:
!IF [50.2#1-10,25,30-35,99]
!IF [50-51#1-10,25,30-35,99]
```

Both these examples say if the data field in columns 50 through 51 has a number from 1-10 or 25 or 30-35 or 99, to execute the question.

2.6.3 Complex IF Statements

More complex IF statements are built by joining two or more simple conditions or arithmetic statements with logical operators, and separating the logical operations with levels of parentheses.

The logical operators and their meanings are:

Operator	**Description**
AND	the conditions on both sides of the operator are true
OR	At least one of the two conditions are true

You can combine the logical operators in any way, provided you use parentheses around the references that belong to each operator. Here is a sample IF statement using a logical operator:

```
EX:
!IF BRAND(3) and AGE > 21
```

A response to question BRAND of 3 and a response to question AGE that is greater than 21 would cause the question to execute. Here is an example using parentheses control for additional levels of logic:

```
EX:
!IF (BRAND(3) AND AGE > 21) OR SHOP$ = "TODAY"
```

A response to question BRAND of 3 and a response to question AGE that is greater than 21, OR a response to question SHOP of TODAY would trigger this condition.

You can combine multiple-label, data location, and/or question number references in one statement.

```
EX:
!IF [1/10$] = "M" OR AGE > 23
```

The above example would be true if column 10 equals M or the response to AGE is greater than 23.

You can also use numeric or arithmetic operators or joiners in a condition statements. (See *3.1.3 DATA GENERATING STATEMENTS, EXPRESSION* Statement for more information.)

```
EX:
!IF (5 + [10.2]) / AGE > 23
```

This says that IF 5 plus the number in columns 10 and 11 divided by AGE is greater than 23, execute the statement.

2.6.4 Using Functions in Condition Statements

Functions are procedures that return a value after performing some special operation. These functions can be used anywhere within a condition or expression (EXP) statement.

The function syntax is:

```
function name (argument)
```

The function name comes first, followed by a left parenthesis, the argument(s), and a right parenthesis. Note that the function name must be directly before the left parenthesis, e.g., X(AGE).

Some functions are called “Extended Functions” and use XF() around the function name. This is for CfMC purposes only (we ran out of room to store more function names so made a ‘sub-function’ capability to add more functions).

Data-Generating Functions

- **IFTHEN, IFTHENELSE**

The IFTHEN and IFTHENELSE functions are used to return numeric values and expressions based on some condition. Both are generally only used with expression (EXP) statements.

Here is the syntax for the IFTHEN function and an example of its use for the return of numeric values:

```
IFTHEN(logical expression,numeric expression)

EX:
!EXPR,,,IFTHEN(QQ5>10,50)
```

A logical expression is any expression that can be resolved as either true or false; a numeric expression results in some numeric value. IFTHEN() returns the value of the numeric expression if the logical expression is true; otherwise, zero is the result. In the example, the result will be 50 if the response to Question 5 is greater than 10; if the response to Question 5 is less than or equal to 10, or missing, the result will be zero.

The IFTHENELSE function returns the value of expression1 if the logical expression is true; otherwise, the value of expression2 is returned. The syntax and an example are given below:

```
IFTHENELSE(logical expression,numeric expr1,
numeric expr2)

EX:
!IF IFTHENELSE(QQ5>10,50,75)
```

In the example, the result will be 50 if Question 5 had a response greater than 10. If the response to Question 5 was less than or equal to 10, or missing, the result would be 75.

In addition to numeric values, the "IFTHEN()" and "IFTHENELSE()" will let you do more complex conditions using question labels within the function.

```
EX:
{XX .2 !EXPR,,,ifthenelse(Q5(1),Q6*2,3}
```

This example indicates that if Q5 is a response code of 1, then multiply the numeric value of Q6 by 2 and put the resulting value in field XX. And, if your Q5 was not a response code 1, then put the value of 3 in the field XX.

For example, the ELSE portion of the function can be used to assign nonzero values to special codes such, as Don't Know or Refused.

- **LOCALSCRATCH**(start, length)

Returns the contents or a region of the local scratch area, a memory area available during a particular Survent session. You must refer to it using the syntax for string or literal data.

```
EX:
!IF LOCALSCR(1,2)="34"
```

If the data in the local scratch area in column one for the length of two is equal to the literal 34 then execute the question. Data gets into and out of the local scratch area by using SPC subtypes K and L. (See *3.1.5 SYSTEM INFORMATION STATEMENTS, Special Information Statements.)*

```
EX:
!IF LOCALSCR(3,4)=[1/1.4$]
```

This example is comparing what's in the local scratch area columns 3-6 (three for a length of four) with what's in the data in columns 1-4.

- **MAX(**numeric expression 1,numeric expression 2, numeric expression n)

Returns the highest number from the numbers specified. The numbers specified can be labels, location references, absolute numbers, QUOTA() functions, etc.

```
EX:
!IF MAX(23,[10.3],AGE) > MIN(QQ1,QQ2)
```

This would cause question execution if the maximum value of the first group of references was greater than the minimum value of the second group.

- **MIN**(numeric expression 1,numeric expression 2, numeric expression n)

Returns the lowest number from the numbers specified. See MAX above for example.

- **RANDOM(#)**

Returns a random number between zero and the number specified. This can be used as a rotate controller (See *3.2.1 INTERVIEW CONTROL COMMANDS, Rotating Questions*). It is usually used only with expression (EXP) statements, not IF conditions.

```
EX:
!IF Random(9)+1 > 3
```

The statement would execute if the random number returned plus one is greater than three; in other words, approximately 70% of the interviews would execute this statement.

Notice that you can control the range of the numbers returned by doing an arithmetic operation. In the example above, a number from one to ten was being returned. The number in parentheses controls the high point of the initial range (0-9) and the +1 changes the range (1-10).

The random function will take as an argument a numeric expression, such as.

```
EX:
!IF Random(Qn23-1)
!IF Random(Qn23)
!IF Random(99)
```

- **X**(label or location)

Returns a zero if the question's response is blank or non-numeric, otherwise the number (numeric value) at the location specified. This is useful for questions that have skip patterns around them, but must still be used in calculations or comparisons.

```
EX:
!IF X([10.2]) + X([AGE]) > 33
```

If the data in columns 10 and 11 plus AGE is greater than 33, execute the statement. If one of the items is blank, it is treated like a zero, and the test can still be done.

If used on a CAT question, X will return the numeric punch value (if single punch) or zero (if blank or multi-punch). If used on a FLD question, X will return the numeric value of the response code(s), if any, or zero (if blank). Multiple responses will show up in one of the following ways:

1 If two responses allowed and two are given, the combined value will be used.

```
EX:
Responses 01 and 02 will return 102
```

2 If fewer responses given than allowed, 0 will be returned.

```
EX:
Response 05 with the next set of columns blank
("05 " )will return 0.
```

If the responses are non-numeric, 0 will be returned.

Response-Checking and -Counting Functions

- **CHECKTEXT**(TEX question label or location)

Returns the number of characters entered as a response to a TEX type question. If the question was not asked, a -1 is returned. If the question was asked but not answered (TEX,B), a 0 is returned. If it was answered, 1-2000 is returned. This is used to determine whether to edit the question later (if many characters), or just to see if a response was given.

```
EX:
!IF CHECKTEXT(OthCars) > 0
```

- **NOT**(condition**)**
- Returns true or false depending on the condition. Returns the opposite of the result of the condition. The condition can be any number of valid conditional statements.

```
EX:
!IF NOT((AGE > 34) AND SEX(M))
```

This executes the statement if AGE is *not* greater than 34 and SEX is *not* M.

- **NUMITEMS(**label or **[**location**])**

Returns the number of “punches” in a label or location. It can be used to count responses in a CATEGORY question. Use the XF(Number_of_Responses()) function to count responses to a CATEGORY OR FIELD question.

- **XF(MATCH_TEXT(**TEX or VAR label or location, string 1 , string 2, string n**))**

Returns the position of a string or set of strings in the answer to long or short open-end questions. It returns the position of the first string if it finds that, otherwise the position of the second string, etc. If more than one string is present, it returns the position of the first string even if it is at a later position than the second. Use this for automated coding applications or to ask follow-up questions based on references to a client product, for instance. You could use multiple strings to check other spellings of the same product. The maximum string length is 80 characters.

- **XF(NUMBER_OF_RESPONSES(**label**))**

Returns the number of responses in a CAT or FLD variable.

```
EX:
{BEV1:
Which of the following beverages did you drink in
the past 24 hours?
!FLD,,3
01 Coca-Cola
02 Pepsi
03 7-up
(-,SKIP POSTFAV) 06 None }
{FAV:
!IF XF(NUMRESP(BEV1)) > 1
Which beverage did you enjoy the most?
!FLD,I,1,IL=BEV1
[SAMEAS BEV1] }
```

FAV will only be asked if more than one response was given to BEV1. This function will count response codes whether they are numeric, alphabetic, or a combination of both.

NOTE: If you want to count the number of PUNCHES in a data location, use the NUMITEMS([location.length]) function

- **XF**(TIMES_ASKED)

Adds 1 to a number every time you go over a question with this function. If you go over it once, back up, go over it again, it returns a "2". ***NOTE***: Other CfMC statements are cleared when you back over them.

```
EX:
{Times: .2 !expr,,,Xf(Times_asked)}
```

Quota-Related Functions

- **MODQUOTA**(quotaname)

Returns the change in the named quota (on this case, as held in memory) since the beginning of the interview.

```
EX:
!IF MODQUOTA(MALES) > 1
```

This example executes if the quota for MALES had been updated more than once (by having multiple quota incrementing statements) since the beginning of the interview. This will not be true for quota updates done using the “NOW” parameter (eg. !QUO,1,name,+1,NOW). In this case the quota file is updated directly, but not the local quota value.

- **MODQUOTN(**quota number**)**

Returns the change in the numbered quota (on this case, as held in memory) since the beginning of the interview.

```
EX:
!IF MODQUOTN(456) > 1
```

- **QUOTA**(quotaname)

Returns the value at the beginning of the interview of the named quota (see *3.1.5 SYSTEM INFORMATION STATEMENTS, Quota Incrementing Statements*), or at the time last read with a READ_NAMED_QUOTAS compiler command (see *3.2.1 INTERVIEW CONTROL COMMANDS*).

```
EX:
!IF QUOTA(MALES) >= QUOTA(MALES.T)
```

This would execute if the quota for MALES was greater than or equal to the target quota MALES.T.

- **XF(QUOTA(<[**data label or location**]** or <quotaname>**))**

This returns the quota value of a quota whose name has been specified in a prior question or location. Here is an example that checks a quota from the sample file vs. it's target quota:

```
EX:
{Quoname: !phone,g,51,8}
{Targname:
\|Quoname|.T
!spc,9}
{!if xf(quota([Quoname])) >= xf(quota([Targname]))
Over quoa for quota \|Quoname, thank the
respondent and terminate
!disp}
```

Note that you can also use this function in the same manner as the QUOTA { } function. That is, if you just put the label of a quota in, it will check that particular quota. If you put a reference in brackets ([ ]s), it knows to check the data location of the specified location or label for the name of the quota to check the value of.

- **QUOTN**(quota number, data location or label)

Returns the current value of the numbered quota specified. The numbered quota can be referenced specifically, or as a number stored in a question label or data location (See *3.1.5 SYSTEM INFORMATION STATEMENTS, Quota Incrementing Statements*).

```
EX:
!IF QUOTN([123]) > 82
```

This says if the quota number stored in column 123 has a value greater than 82, execute the statement. This may slow down the system if used repeatedly.

- **QUOTANOW(**quotaname**)**

Returns the current value of the specified quota. Using this repeatedly may slow down the system. Consider using the READ_NAMED_QUOTAS compiler command instead.

Phone File-Related Functions

- **FONESTATUS**( )

The FONESTATUS function is used to return the current status of the phone record. This function is often used with a predictive dialer to see if a status has been returned. An "0" is returned for nonpredictive dialer systems unless a PHO,S statement has set a status.

```
EX:
!IF FONESTATUS()=1
```

- **FONETEXT**(column, width)

Returns the contents of the specified field in the phone file.

```
EX:
!IF FONETEXT(60,3)="ZOO"
```

- **LAST_CALL**()

Says whether or not you are on the "LAST CALL" for the phone number you are calling. By last call we mean that after this call the phone record will be resolved with status 94 because it's maximum number of calls will have been reached if it is a system call. It returns a "0" if it is not the last call, and a "1" if it is. This is used for advanced programming of call statuses based on whether the call is the last one or not.

- **XF(MARKET_WEIGHT(<**marketname or **[<**data label or location**>]>))**

This returns the value of a particular market weight. You can either refer to a particular market using "marketname" or refer to a market name stored in the data using the "[label or location]" syntax.

- **XF(MAX_ATTEMPTS)**

This returns the "Maximum Attempts" setting from the phone file, which is the maximum times you can call a particular phone number. You can then look at this value to determine how to status the current call.

This function allows you to include max_attempts as one of the possible resolved statuses so you can do something based on that criteria (eg. add a new phone record with the !phone,a statement).

This was previously easy to do if the status is a standard resolved status, but, if the number would have been resolved due to maximum attempts, it was not possible.

Note that Survent picks up the fone header at the start of the interviewing session, so the value is whatever it was at the session start.

- **XF(NUMBER_REDIAL)**

This returns how many times the number has been re-dialed during this interviewing session either using the “REDIAL” command or automatically.

Date-Time Functions

- **XF(DATE_TIME_DIFFERENCE(** date1, date2, type

returned **))**

This function returns the difference in 2 date/time strings in years, months, weeks, days, hours, minutes, or seconds. It reads two dates and a keyword and then returns a value. The dates can either be actual dates, or they can be question labels or [location.length] references where the dates are stored.

The date strings must be dates in the format YYYYMMDDHHMMSS. This is the same format returned by the !SPC,3 statement. The string must have as many characters as the keyword requires, for instance, if you want to return seconds, you need all 14 characters, but to return days you only need 8 characters.

The <Keyword> is one of SEConds, MINutes,HouRs,DAYs, WeeKs, MONths,or YeaRs. The value returned is the difference from Datetime1 to Datetime2. If Datetime2 is more recent than Datetime1, a negative value is returned. If the difference is 59 minutes and 59 seconds, it returns 0 for the number of hours and 59 for the number of minutes. If you are looking for rounding, you will need to use the next keyword and see if its value is over half of what is needed to kick the next value (i.e. >= 30 minutes) and add 1 to the value if true.

Datetimes are only compared on the shorter of the two values. For instance, if Datetime1 has only a date and Datetime2 has both a date and time, then only the days/months/years values will be nonzero, regardless of the time on Datetime2. Hours, minutes, and seconds all return "missing" in this case.

If either string has more than 14 characters, extra characters are ignored. If either string has less than 4 characters this is a spec error.

```
EX:
XF(DATE_TIME_DIFF( born, today, DAYS))
```

This would return the number of days since the born date to today’s date.

- **XF(DATE_TIME_OFFSET(** place to save date, date to compare, type of return, # to offset by **))**

This function return a dates in the format YYYYMMDDHHMMSS given a starting date (YYYYMMDDHHMMSS) string and the amount to offset it by in years, months, weeks, days, hours, minutes, or seconds. The adjustment value can be positive or negative depending on whether you want to go forward or backwards in time.

You may use a datetime string that is shorter than 14 characters, but you must give enough characters in the string to match the keyword, for instance, if you add hours, your starting date must include hours.

```
EX:
XF(DATE_TIME_OFFSET(newdate,olddate,weeks,15))
```

This would return a date 15 weeks after the date in the question labeled “olddate” into the question labeled “newdate”.

- **XF(DAY_OF_WEEK(** date **))**

This function returns the day of the week associated with the date specified. "date" can be an actual date, or it can be a label or [location.width] where the date is stored. You must have at least YYYYMMDD in the date field, but can use up to a 14 character datetime string in the format YYYYMMDDHHMMSS. For example:

```
EX:
{DATE: .8
20040121 ''Put date in data
!SPC,9}
{DOW: .1 ''Record the day of the week
!EXPR,,,XF(DAY_OF_WEEK(DATE))}
```

This would return a "3" for Wednesday. Monday is 1, Sunday is 7.

- **TIMEDIFF** ([loc1.12$] or label$,[loc2.12$] or label$)

Returns the difference of two times in seconds (V7.6l and 7.7). Times must be in the format YYYYMMDDHHMM for version 7.6L and 7.7 . Here is an example:

```
EX:
!IF TIMEDIFF([10.12$],[30.12$]) > 60
```

This states that if there is more than one hour between the time specified in location 10.12 and the time specified in columns 30.12, execute the statement.

NOTE: V7.6L and 7.7 read yyyymmddhhmmss or anything less. Also, the first time specified should be the later time, while the second time should be the earlier time (TIMEDIFF subtracts TIME2 from TIME1).

The maximum value returned from TIMEDIFF is 10,000,000 minutes (20 years). In versions 7.6L and 7.7, the four-digit year format of YYYYMMDDHHMM as well as the old two-digit format of MMDDYYHHMM are supported. In the two-digit format, years less than 28 are considered 2000-2027, while years greater than 27 are 2028-2999.

Also in version 7.6l, the function returns seconds. Use an EXP to divide the result by 60 to get minutes or use dump switch >dump S7.

Mode-Checking Functions

- **DATAGEN**( )

Returns a “1” or “true” if the interview is using the random data generation operation (see *2.7.2 RANDOM DATA GENERATION*), “0” if not. This is used to skip questions (e.g., screener questions) or other things that random data generation would not work well with. There are no arguments between the ( )s.

```
EX:
{
!IF DATAGEN()
!GOTO, FIRST}
```

This would go to the question FIRST if random data generation was being done, otherwise continue to the next question.

- **DIALER**( )

Returns “1” if the study is using a predictive dialer, “2” if it is using a power dialer (MSG dialer only, and “0” (“false”) if not. This is used in cases where users have alternate code for the study with different dialer types or without a dialer.

- **XF(DIALER_TYPE)**

Returns “"0" if no dialer present, "1" if SER dialer, "2" if PRO-T-S dialer, "3" iff NOBLE dialer, and "4" if Stratasoft dialer. This is used in cases where users have alternate code for the study with different dialer types or without a dialer. Other dialer types will be added as supported.

- **SOUND_SERVER**( )

Returns “1” (“true”) if a CfMC sound server is active and “0” (“false”) if it is not.

- **WEB_SURVENT**( )

WEB_SURVENT( )

The webSurvent() function returns codes specifying what mode you are in and why you are being suspended. The return codes are:

1 Running in webSurvent mode

2 Suspended in webSurvent or webCATI

4 Was suspended due to a suspend timeout (also known as "autosuspend", where the survey was suspended because no response was given in the last 30 minutes or so) in webSurvent or webCATI

8 Running under webCATI

16 Running as a web utility (eg. "&&&phrpt.qff" to run PHONERPT under Mentor)

32 Suspended because there was a webSurvent or webCATI killed session

For instance, you can use !IF websurvent=1 in the !SUSPEND block to control any features you have that should only be invoked when suspended by an interviewer and NOT when auto-suspended.

NOTE: For more examples that use these functions, see *3.1.3 DATA GENERATING STATEMENTS,* EXPRESSION Statement.

System-Related Functions

- **XF(PROCESS_ID)** (UNIX)

Returns the process ID of the current Survent session. This can be used for accounting or to remove process IDs that are not active. You can use this with !IF conditions and !EXPR statements. You can use XF(PID) for short.

- **XF(USABLE_STUDY**(questionnaire filename)**)**

Returns a “1” if the questionnaire file is present, and a “0” if not. This is usually used when you have a list of studies for the interviewer to choose from and have !CALL statements to execute the questionnaire if it exists.

2.7 TESTING THE QUESTIONNAIRE

Just because your questionnaire compiles, that doesn't mean you're ready to start interviewing. There are several things you need to do to ensure a good questionnaire.

2.7.1 Testing Checklist

- **Check your specs**

Look back over your specs. Recheck your logic, look for typos and inconsistencies. Checking these items on a printed listing makes it easier to find problems than looking at one screen full of your specs at a time on your terminal. You can also check off items as you verify their correctness. If you are working in column-free mode, hardcode the data locations. If using TEX questions, make sure you have defined a TEXT_START on your header statement, and that there is a buffer of blank columns between the last non-TEXT column used and the TEXT area.

- **Compile**

If your run doesn't compile successfully, you have an obvious place to devote your efforts. But even if you have a good compile there are still things to check. Look at all the messages and warnings that PREPARE produces. If you don't understand one, spend time on it until you do. Whenever PREPARE prints a warning, it is really saying "I hope you know what you're doing" or "You really should have done this another way, but I'll help you out."

- **Screen check**

Run Survent and look at the screens. Are they readable? Are they consistent? Are skips flowing correctly? Is there any text that doesn't stay on the screen long enough for the interviewer to read it?

- **Data column usage**

You also need to think about whether the data is correct. One consideration is whether you've used the same data location for more than one question. You can easily find this out by checking the CHK listing.

- **Random Data Generation**

Another concern with the data is whether you've written any questions that never get asked because of bad skip patterns. It's difficult to check every skip pattern yourself. See the next section, Random Data Generation, for a way to help you. You could run HOLE or FREQ on the data file generated and look for blank columns.

Final thoughts

As you can see, there is some overlap in the checking done in the items listed above. All these steps do not have to be done, nor do they have to be done in the order listed. But you do need to do something that accomplishes the purpose of each item.

Once you are more experienced with the program, you will have a better idea of what steps you need to take. The most important thing to remember is to *always test your questionnaire*! For best results, have many people test the questionnaire – even after what seem to be small changes. It's better to be safe than sorry.

2.7.2 Random Data Generation

This option is used by testers to generate responses to questions and make sure that all the necessary skip patterns are being followed, or to produce demonstration data sets or reports. This can be used on any questionnaire.

Use the command RDG and its associated options to turn on random data generation.

The syntax for RDG is:

```
    RDG option,option
```

Here are the options you can choose from. They can be specified in any order and are separated by a comma or space(s). If no options are specified, you will be prompted for what you want, with the prompt showing the default in parentheses.

Option	Description
SHOWANSWERSONLY	Display answers only, and no question text.
DEMO	Waits for a keystroke after each answer is shown.
NCASES=#	Number of cases to initiate.
PAUSE=#	The delay after each answer, specified in seconds.
SEED=#	Allows you to reproduce the same set of answers at another time.
BACKUP	Backup occasionally in the questionnaire.
SHOWQUESTIONS	Displays questions and answers on the screen.

NOTE: You must have debug capability (DBUG interviewer ID or D in EMPLOYEE.XXX file) (see *4.4. INTERVIEWING WITH SURVENT, Employee Information File,* for more information on this) in order to use the RDG option. This is so interviewers will not mistakenly use this option.

Putting >DUMP S1 in the INITIAL file allows RDG mode for any interviewer. This is not suggested.

Putting >DUMP S2 in the INITIAL file will also allow RDG and Debug mode for any interviewer. This is also not suggested. Either of these commands typed at the in-between interview prompts will turn on this mode at that time, even for interviewers without debug capability.

Specifying RDG HELP at the in-between interview prompt will bring up a small help screen.

Here is a sample procedure:

- Enter SURVENT to load the program.
- At the **Return** to interview prompt enter:

```
RDG (plus other options you want to set)
Enter -->
```

The interviews will begin and the program will place you at the Survent interview prompt when the interviews are completed. You could then use the utilities to produce reports, and check the counts for proper values, before starting actual interviewing. If you run HOLE, look for blank fields; they may indicate questions that are never asked.

By setting a seed for random data generation you will be able to reproduce the same set of answers in another run.

If a Survent blow error occurs, data generation will stop, and a separate data file will be produced containing the error. The data file will be put in a <study>.B_ subdirectory. A message will also print explaining the error and the question it occurred on. To debug the problem, you can use Survent's View option on the data file containing the error up to the point of the problem, and then fix the problem in the PREPARE specification file. If a Survent blow error occurs during random data generation, you will also see the seed of the current case. If after changing your specs you would like to test the blown case, start up again and specify the seed reported earlier by the program.

The number of interviews you actually get in your data file may be less than the number you asked for if you are aborting

(SPC,B) interviews in your specs. Also see *2.6.4 USING FUNCTIONS IN CONDITION STATEMENTS, Conditional Functions* for information on the DATAGEN function, which will exclude blocks of questions from being randomly generated. VAR questions will have the alphabet (and, if enough characters are needed, punctuation characters) entered in the data—if you see the alphabet showing up in your real data, you may have forgotten to purge (or rename) your data file before you started collecting real data. "One line of randomly generated text" is generated for each TEX question that is asked. You will see this line in the data file when TEX question locations are displayed with the DT command in CLEANER or the LIST utility program.

If you want to stop RDG while it is still running, press **Ctrl-Y** (**Ctrl-BREAK** in UNIX). You will be prompted as to what you want to do. Your answer choices are:

RDG to continue

RDG # to do # more questions in the current interview, then stop and wait for you to tell it what to do next

Q to quit

Weighting Responses

This option allows you to weight each response in a CAT or FLD question to control how often the random data generator will choose a particular response. Since a weight must be specified for each item in the response list, this option is recommended for questions with short lists. This option would most likely be used to get the random data generator through the screener portion of the interview.

The syntax for weighting responses is:

```
!RDG #,#,...,#
```

where:

- Each # must be separated by a comma.
- Weights must be a positive number from 0 - 99.

- The number of weights must be equal to the number of responses.
- Response weights must add up to 100.

NOTE: !RDG can come before or after any !IF condition statement, but must come after the question label line and before the question text.

```
EX:
{CANDY:
!IF AGE>18
!RDG 90,9,1
Have you or any member of your family purchased
candy during the last month?
!CAT
01 Yes, has purchased
02 No, has not purchased
03 Refused}
```

In this example we have weighted the responses such that the Yes response will be chosen approximately 90% of the time, the No response 9%, while Refused would rarely be chosen. In this case a Yes response is needed to continue in the questionnaire.

Skipping Questions

Should you want to skip certain questions when generating random data, you can use the DATAGEN function. Using this function on an IF condition attached to a GOTO question would let you skip around screener questions, quota checks, or other question types that you choose to treat differently. (See *2.6.4 USING FUNCTIONS IN CONDITION STATEMENTS* for more information on the DATAGEN function.)

Multiple Response Questions

The random data generator employs a formula to determine how often it will generate more than one response on multiple-response CAT and FLD questions. You can weight the responses so that one or more would be chosen more often when multiple responses are generated.

```
EX:
{CARDS:
!RDG 70,10,10,9,1
Which of these credit cards do you own?
!CAT,,4
1 General Purpose Cards
2 Bank Cards
3 Retail Store Cards
4 Gas/oil Company Cards
(-) X none
}
```

In this example, GENERAL PURPOSE CARDS is weighted to be chosen approximately 70% of the time. This means it would account for about 70% of the total responses given to this question. Response NONE would be a rare choice (1% of the time).

Interviewer Testing with STARTRGD and RDGDELAY

The commands "STARTRDG" and "RDGDELAY" can be used to start "background" interviewers in Random Data Generation mode at different speeds on different studies. This enables you to do full "load" testing on your system without having to start up individual stations.

To use these commands, first specify "RdgDelay ##" where ## is the number of seconds between each question. The default is 20 seconds. Then specify "startrdg ##-## " to start interviewers; note that this uses the same syntax as the "start" command.

Smart Random Data Generation

"!Smart_RDG" commands are an extension of the Survent RDG (Random Data Generation) function. They allow you to check answers across multiple questions and force them to have particular responses (if random filling of the questions does not return the value you need) so the RDG run can continue.

!Smart_ RDG is a compiler directive that can be used outside of a question. RDG mode previously allowed no control of the responses across multiple questions, just the limits for a particular question. For example, if you were trying to continue only if the sum of a set of questions was 100 (eg. adds to 100%), it could seldom generate the data such that the values added to 100. This can lead to an infinite loop during the RDG run and cause the run to fail.

!Smart_RDG commands first attempt random data entry for the number of tries specified, then fills the value(s) based on your criteria.

WebSurvent and WebCATI will not execute !Smart_RDG statements. So, even if you are programming a Web survey, you must use terminal mode Survent when testing using RDG and !Smart_RDG. This can be done easily enough using "netsurv" to execute the interview (Netsurv is also covered in this chapter).

NOTE: If you prefer, you can use "!RDG" as shorthand for "!Smart_RDG".

Programming issues

!Smart_RDG commands require !RESET statements be used for backend checks instead of !GOTO statements, as !GOTO does not clear previous data, a requirement for !Smart_RDG. However, most programmers doing Web surveys use !GOTO functions for backend checks, as they are more user-friendly for respondents (previous answers are maintained).

To accommodate both !Smart_RDG and a user-friendly program design, you can use a couple of techniques. The first involves using conditional DATAGEN() functions so that the !RESET will be executed in RDG mode and a !GOTO in standard interviewing mode, for example:

```
      {!IF DATAGEN() AND (X(Q1A)+X(Q1B)+X(Q1C)<>100) ''An RDG interview
      !RESET PREVQ }
      {!IF NOT(DATAGEN() AND (X(Q1A)+X(Q1B)+X(Q1C)<>100) ''A live
interview  !GOTO PREVQ }
```

The second technique is to use an >IF_DEFINE statement to choose between the !GOTOs and !RESETs at compile time:

```
{!IF (X(Q1A)+X(Q1B)+X(Q1C)<>100)
>IF_DEFINE @RDG_RUN
!RESET PREVQ }
>ELSE
!GOTO PREVQ }
>ENDIF
```

Note in this case you need a command like ">define @RDG_RUN" at the top of the questionnaire whenever you compile a version using RDG, and would have to comment that statement out (''>define @RDG_RUN") and re-compile for live interviewing.

Syntax description

!Smart_RDG: This starts a Smart_RDG block and is in effect until another !Smart_RDG statement is encountered. The syntax of a Smart_RDG statement is:

```
     SYNTAX:
{!Smart_RDG Keyword(<Value>,Question List) <TRIES=##>}
{!Smart_RDG Keyword(Question List)=Value <TRIES=##>}
{!Smart_RDG FREE}
```

!Smart_RDG FREE: This disables or resets previous !Smart_RDG statements. It is recommended that you use !Smart_RDG FREE after each Smart_RDG function has finished.

!Smart_RDG Keyword(): Eachs keyword is specific to a type of check commonly used in Survent. For example, a check that requires multiple numeric (!NUM) questions to equal a specific number would require the CONSTANT_SUM() keyword.

Question List: This is list of question labels which will be used with the assigned keyword in order to meet the necessary conditions. There is a 50 label limit, but asterisks (*) can be used as wildcards for questions with like-named labels.

```
     EXAMPLES:
{!Smart_RDG Keyword(QLBL1,QLBL2,QLBL3)}
```

```
{!Smart_RDG Keyword(QLBL*)}
```

The first example would use labels qlbl1, qlbl2, and qlbl3. The second example would use all labels that begin with "qlbl".

TRIES: This is the number of random attempts to make before the Smart_RDG statement sets the value as specified.

```
     EXAMPLE:
{!Smart_RDG CONSTANT_SUM(QL0*)=100 TRIES=10}
```

In this example, SmartRDG will enter random data in the QL0 series of questions ten times. After the tenth attempt, the CONSTANT_SUM keyword will be implemented, and the data generator will answer the QL0 series so that the questions in the QUESTION LIST add to 100.

If TRIES is not specified, SmartRDG immediately generates the data to match the keyword criteria.

Keyword functionality

CONSTANT_SUM: This keyword checks that a group of numeric (!NUM) questions add up to a certain value.

```
           SYNTAX:
!Smart_RDG CONSTANT_SUM(Question List)=<Label or ##> TRIES=##
           EXAMPLE:
!Smart_RDG CONSTANT_SUM(Q1,Q2,Q3)=100 TRIES=10
!Smart_RDG CONSTANT_SUM(Q1,Q2,Q3)=TOTAL TRIES=10
```

After the specified number of TRIES using random filling of the questions, it will force these !NUMs to add to value assigned. Notice the value to assign can be either a specific number or the value of a previous label.

RANK: This keyword deals with checks that require a unique rank or code across all available !FLD or !NUM questions. If there are duplicates it should be an error in this case.

```
     SYNTAX:
!Smart_RDG RANK(Question List) TRIES=##
```

```
         EXAMPLE:
   !Smart_RDG RANK(Q1,Q2,Q3) TRIES=10
```

The above tries to get unique values randomly, then after # tries it supplies a unique set of values. For numeric variables (!NUMs), it assigns values starting with 1 and increasing by 1. If used with a !FLD code list it supplies a unique code for each.

RANKTOP: This deals with checks that require a unique rank or code for some, but not all of the !FLD or !NUM questions. It is useful for rank questions with a limited number of ranks out of a set (top 3, bottom 3, etc.)

```
            SYNTAX:
!Smart_RDG RANKTOP(<# to rank>,<Question List>), TRIES=##

            EXAMPLE:
!Smart_RDG RANKTOP(3,Q1,Q2,Q3,Q4,Q5), TRIES=5
```

Will randomly insert unique responses between 1 and the # TO RANK across !NUM or !FLD questions in QUESTION LIST, and leave others blank. Questions or grids must use B subtype to allow for blanks in data.

RANK_WITH_DONT_KNOW: This keyword will force unique rankings for each question, but also allows for an non-unique exception code. This function assumes the last response item in the response table in the exception, unless Don't Know Other Response (DKOR) is specified, as per Example 3 and 4:

```
                SYNTAX:
!Smart_RDG RANK_WITH_DONT_KNOW(Question List) <DKOR=xx> TRIES=##
                EXAMPLES:
!Smart_RDG RANK_WITH_DONT_KNOW(Q1,Q2,Q3,Q4) TRIES=5
!Smart_RDG RANK_WITH_DONT_KNOW(Q1,Q2,Q3,Q4), DKOR=99, TRIES=10}
```

Supplies a unique rank for each question in the QUESTION LIST, and allow one non-unique exception code for the last response item in the !FLD.

ONLY_ANSWER: This keyword will force RDG to enter a specific answer for all specified questions.

```
SYNTAX:
        !Smart_RDG ONLY_ANSWER(Answer,Question List) TRIES=##)

             EXAMPLE:
     !Smart_RDG ONLY_ANSWER(01,Q5,Q7) TRIES=5)
```

The answer can be a number or a response code.

OTHER: A keyword to always meet Other Specify criteria in a !FLD question. When used, this will always select the other response code as an answer choice, so 'checktext' backends are always met.

```
              SYNTAX:
!Smart_RDG OTHER(Question Label), OTHER_RESP=x, TRIES=#
              EXAMPLE:
!Smart_RDG OTHER(Q1), OTHER_RESP=99, TRIES=5
```

Since RDG will always fill a !VAR or !TEX question with data unless it has a condition, this function will also always select the other response code, so back-end checks that enforce the !FLD to !VAR/!TEX relationship are met.

2.7.3 Tracing Problems

USING THE TRACING DUMP SWITCHES

To trace interview data or track test conditions, use DUMP:n#" on the Survent command line, or ">DUMP n#" inside Survent if you are in Debug mode. This will enable you to use the following interview tracing features:

- **n1** - The label and number of the next question is shown and it tells you whether it will execute the next question based on the condition. This allows you to check skip patterns.
- **n2** - This halts the display so you can see dump information (but not wanted if in random data generation hunting bugs)
- **n7** - This prompts you for a set of columns to display, then shows you the data for those columns over and over until you turn it off or pick another set of columns to display.

- **n8** - This shows the data for each question including its data position and width before the question is asked.

Use >DUMP n to get all tracing features turned on, and >DUMP –n to turn them off.

LOGGING INTERVIEWER RESPONSES

Sometimes when tracing problems, it is useful to get an ASCII log of all the interviewer commands. Use the LOG command at the <Return to Interview> prompt in Survent to do this.

In some cases you need to save the logged responses immediately after each question. This is particularly true if the program is blowing up and you don't know why. In this case, you may use the LOGDEBUG command tells Survent to save the log file after every question.

When the questionnaire blows up, you can read the log file to see what may have caused the problem. The file is save as LOG<intv_id> in the CFMC IPCFILES directory.

Note that when a questionnaire gets a BLOW error, it automatically logs that interview to the interviewer's LOG file, even if logging was not turned on. This is one place you can always go to see what responses were entered and try to fix the problem.

FIXING BLOW ERRORS AND VIEWING BLOW FILES

Sometimes your questionnaire design is incorrect such that certain response produce a BLOW error which causes the interview to be aborted and the data to be saved in an alternate data file in the BLOW file directory. The program will note the ERROR #, a text message, the question where the error occurred, and the name of the file it saved.

To get more information on what the error means, see Appendix D. The most typical blow error is ERROR # 108 where you use CAT or FLD,A subtypes that have not properly had their data filled in before being executed.

It is useful to review the responses to see what may have caused the BLOW error. You can do this by VIEWing the BLOW file. How to do this is slightly different on each system:

- **DOS** - Specify the name of the data file to view on the configuration screen; you must be in the blow file directory for the study (usually \cfmc\data\<study>.b_).
- **UNIX** - Specify <study>,BLOW=<filename> at the "Type questionnaire filename" prompt. The program will find the file in the blow directory for the study no matter where you are logged in if the CFMCDATA variable is set.

You will then be placed in VIEW mode on that file. Walk through the questionnaire to the place it blew up. If you'd like, you can turn on the >DUMP n switches to see the results of conditions leading to the BLOW to help trace the problem.

2.8 ADDING YOUR OWN HELP MESSAGES

If you want to define text to be displayed on the bottom of the screen when the question executes that will perhaps clarify that question, you can put a !HELP line in your question block. It is specified at the top of the question block before the question text. This can be especially useful inside of a GRID; when you move from question to question the text at the bottom of the screen will change accordingly.

This will override any messages from the HELP_<question type> compiler commands.

```
EX:
{Q1:
!HELP Enter a value from 1-10, DK or NA
How many children are in your household
!NUM,,,1-10,,DK,NA}
```

Note also that you can back-reference responses to prior questions on the !HELPline:

```
EX:
{Q2:
!HELP Children have: \:Q1:
How many of those children are less than 5?
!NUM,,,1-Q1,,DK,NA}
```

This would display the answer to Q1 on the help line at the bottom of the screen.

2.9 MAKING CHANGES ONCE A JOB HAS STARTED

2.9.1 What to Do When a Study Has Been Soft Coded

What should you do when a study has been soft coded (data locations not specified) and you are ready to go live and need to make sure that changes made during the interviewing process do not greatly affect the data layout?

There are basically three different approaches. The one you will want to use will depend on how many changes and what kind of changes you expect to make after the study has gone live. On a job that will run for only two or three nights, you might want to use approach #2. For a wave study, you might want to use approach #3.

APPROACH NO. 1

You expect some amount of changes. This is the standard recommended method.

A. Issues

1. Positives

- You can determine the data position for any existing question from the spec file.
- You can easily modify any existing data question because default values have all been filled in.
- You are less likely to make a data error.

2. Negatives

- Repeats are expanded, so the file is much larger and repeat blocks are difficult to modify.
- Default values are all filled in, so each question has unnecessary information in it.

B. When Going Live

Make sure the top of your program has the HARDCODE compiler command in it right after the Header statement and before your first question block. Then, after you have done the final compile, but before you go live, follow these six steps:

1 Rename your existing spec file <study>.QPX to something like <study>.ORG.

2 Rename the existing check file <study>.CHK to something like <study>.CHO.

3 Rename your existing qsp file <study>.QSP to <study>.QPX.

4 Edit the file <study>.QPX and make the following modifications:

Add the following lines at the top of the file.

```
~COMMENT
```

Any comments you would like in the file.

```
>PURGE_SAME
~PREPARE COMPILE
```

Add the following line at the bottom of the file.

```
~END
```

Make sure that questions in this file have data locations on the label line. For instance:

```
'44.50 404
{ GENDER: 404
Enter Sex of Respondent
!CAT
() 1 Male
() 2 Female
}
```

It is the 404 on the line with GENDER that is needed.

5 Recompile the new <study>.QPX file.

C. Making Changes After Going Live

1. Adding Questions

If you are adding questions, look at the CHK file and find the last column that was used. Make sure that you put any new questions in UNIQUE data positions after that point in the file. For instance, if the CHK file says the last column used is 575, put the new question in data position 601. This would look something like the following:

```
{ NEWQN: 601
This is an added question
!NUM,,,0-100 }
```

This puts the question called NEWQN in position 601, so it neither shifts nor overwrites any existing data. Be sure to double check the CHK file to make sure the new column(s) assigned do NOT overwrite any existing columns.

2. Removing Questions

If you wish to remove a question from the survey, use the HIDE option on the label line.

```
{ DONTWANT: 273 HIDE
Don't want this question anymore
!VAR }
```

3. Modifying an Existing Question

If you need to modify an existing question and are not increasing the number of data columns it uses, then there is no problem.

If you need to increase the size of an existing question you need to both create a new questionand hide the original one. You must copy the original question and assign the new question a new data position outside the existing data and use the HIDE option on the original question and change its label, so as not to create a duplicate label.

Two unfortunate consequences of this change will be:

- The data for this changed question will need to be netted back together at some point.
- Previously suspended interviews will probably not FIXRESUM.

If there are questions you think are likely to increase in size, then you can put ".n" on the label line of the question in the original file, where "n" is the number of data columns to allocate for that question.

4. Important: Check for overwrites before recompiling

After you have recompiled with your changes, look at the new CHK file and make sure you have no unwanted data overwrites.

APPROACH NO. 2

This is probably best for *minimal* changes.

A. Issues

1. Positives

- You are able to use your original spec file.
- Repeats are not expanded.

2. Negatives

- You cannot determine where the data is stored from your spec file.
- Data overwrite errors are more likely to occur.

B. When Going Live

After doing the final compile, but before going live, you will want to follow the next three steps:

6 Rename the check file <study>.CHK to something like <study>.CHO.

7 Look at the CHK file to determine what the last data column was that was used. If you are not currently using the work area, then set the WORKSTART option in the Header to a column greater than that column. For instance, if the last column used is 575, then set WORK_START=601.

```
[SAMPL,CASE_LENGTH=8000,COMMENT=”Sample
Header”,TEXT_START=2001,&WORK_START=601]
```

8 Recompile the spec file.

C. Making Changes After Going Live

1. Adding Questions

If you are adding questions, just hard code the new question in the work area using the WORK keyword, or place it in a UNIQUE previously unused data column anywhere after the WORK_START column. If WORK_START is 601 and the work area has not been used yet, you can put the first new question starting in data position 601. For example:

```
{ NEWQN: WORK
This is an added question
!NUM,,,0-100 }
```

or

```
{ NEWQN: 601
This is an added question
!NUM,,,0-100 }
```

In both cases, this will put the question called NEWQN in position 601, so it neither shifts nor overwrites any existing data. Be sure to double check the CHK file to make sure the column(s) you assign do NOT overwrite any existing columns.

2. Removing Questions

If you wish to remove a question from the survey, use the HIDE option on the question label line.

```
{ DONTWANT: HIDE
Don't want this question anymore
!VAR }
```

3. Modifying an Existing Question

If you need to modify an existing question without changing the number of data columns it uses, then there is no problem.

If you want to reduce the number of columns the question uses, you need to put ".n" on the label line, so that the question will continue to use "n" number of columns. For example, if the following question had its maximum changed from 100 to 50 you would want to put .3 on the label line so it would continue to use the 3 columns, instead of 2.

```
{CHANGE: .3
Enter in your number here
```

```
    !NUM,,,0-50,,DK }
```

If you want to increase the size of an existing question you need to both create a new question and hide the original one. You must copy the original question and assign the new question a new data position past the WORK_START position and use the HIDE option on the original question and change its label, so as not to create a duplicate label.

Two unfortunate consequences of this change will be:

- The data for this changed question will need to be netted back together at some point.
- Previously suspended interviews will probably not FIXRESUM.

If there are questions you think are likely to increase in size, then you can put ".n" on the label line of the question in the original file, where "n" is the number of data columns to allocate for that question.

4. Important: Check for unwanted overwrites

After you have recompiled with your changes, look at the new CHK file and make sure you have no unwanted data overwrites and compare <study>.CHK with <study>.CHO to make sure all the old questions are being assigned the same data positions.

APPROACH NO. 3

This is probably best for Wave studies where you expect *massive* changes.

A. Issues

1. Positives

- You are able to use your original spec file.
- Repeats are not expanded.
- You can determine where the data is stored from your spec file.

2. Negatives

- It requires extra labor to assign the data columns before you go live.

- Data overwrite errors are more likely.

B. When Going Live

After doing the final compile before going live, you will want to do the following:

1. Copy the existing spec file <study>.QPX to something that resembles <study>.ORG.
2. Rename the existing check file <study>.CHK to something resembles <study>.CHO.
3. Look at the sum file <study>.SUM to determine what data columns the qfile is currently using. Start modifying the spec file so that those data locations are hard coded into the file. This is a relatively long error-prone process, but it does mean that all previous modifications of the file avoided the used of data positions.
4. Put the option DATA_LOCATION_REQUIRED in your Header statement.
5. Recompile with the new spec file.
6. Make sure that the new check file <study>.CHK exactly matches the old one <study>.CHO.

C. Making Changes after Going Live

1. Adding Questions

If you are adding questions, just hard code the new question in a UNIQUE previously unused data column anywhere in the data record.

2. Removing Questions

If you wish to remove a question from the survey, use the HIDE option on the question label line.

```
{ DONTWANT: 421 HIDE
Don't want this question anymore
!VAR }
```

3. Modifying an Existing Question

If you need to modify an existing question and are not increasing the number of data columns it uses, then there is no problem.

If you need to increase the size of an existing question you need to both create a new question and hide the original one. You must copy the original question and assign the new question a new data position outside the existing data. Also use the HIDE option on the original question and change its label, so as not to create a duplicate label.

Two unfortunate consequences of this change will be:

- The data for this changed question will need to be netted back together at some point.
- Previously suspended interviews will probably not FIXRESUM.

If there are questions you think are likely to increase in size, then you can put ".n" on the label line of the question in the original file, where "n" is the number of data columns to allocate for that question.

4. Important: Check for unwanted data overwrites

After you have recompiled with your changes, look at the new CHK file and make sure you have no unwanted data overwrites.

2.9.2 Changing to a New QFF File While a Study is Active

You can only use this approach if the changes you are making do not add any references to new quota names that did not exist in the original. If you are adding quota names you ***must*** bring the study down to make those changes.

1 Copy the current QPX and the live QUO file into a temporary directory/group. Copy/build a FON/FNX file if you need it for testing.

2 Then use the QFFFILE=<newname> option in the Header of the spec file to cause the new QFF file to be created with the name

newname. You would usually use a name that is similar to the original study code with an additional number or letter appended to the end of the name. For instance, if the original Study name was BANK, you might have the new name be BANK1. *Do not change the study name on the header*.

3 Compile your QPX file and test all your changes.

Once you are satisfied with the changes, then copy the new QFF file into the QFF directory. Remember to also copy the QPX file back into a/the permanent directory.

4 The Supervisors then should be able to issue a CHI <station number> <NewQFFname> to any stations that are currently working on that study. As interviewers finish the current interview they are on, they will be switched to the updated QFF file.

5 Once all the interviewers have been switched, then the old QFF file should be deleted, so that the supervisors do not inadvertently start up interviewers on the old QFF file. *Do not shut down the study*. A supervisor could still issue a Start command to the old QFF, but the interviewer will get a fatal error message when they try to start up.

You may or may not want to rename the new QFF file name back to the original name overnight. This depends on what is easier for your supervisors to deal with.

3 ADVANCED PREPARE SPECIFICATIONS

3.1 CONTROL STATEMENTS

There are several question types that are called control statements. These serve many functions necessary for the interviewing process.

The specific syntax, available subtypes and options vary for each control statement and will be described accordingly in their specific sections. Here is a list of the control statement functions and their related question types.

- Control the flow of the interview. These statements cause the interview to go forward or backward based on some condition or previous response.

Goto (GOTO) statements are used to skip to a specified question or as a place to go to.

Reset (RSET) statements return the interview to a previously asked question.

• Screen control/display (no data collected):

Display (DISP) statements are used to put text on the screen (with or without a pause) for the interviewer. No data entry prompt (-->) is displayed. Use this to display information to the interviewer or to conditionally paint the screen for a data entry question.

- Generate data internally. These statements can be used to do calculations during the interview or change the data collected:

Expression (EXP) statements calculate values and return a numeric value or just write a string. You can use question labels, data, arithmetic joiners and functions.

Generate (GEN) statements add or remove responses, move data or blank questions. These control CAT or FLD subtypes A, C, D, and I response lists as well as act on existing data.

- Modify/enhance the function of data entry questions:

Disk-based field (FLD) questions allow you to have an unlimited response list set up in levels, and allow you to modify the list while interviewing is in progress.

Edit (EDIT) statements let interviewers edit text from TEX and VAR questions using a full-screen or line editor.

Loop/Endloop (LOOP/END_LOOP) statements are used to repeat a series of questions for each response to a multiple-response or numeric question.

- Share information with sources outside the interview:

Phone (PHO) statements interact with the phone system to get a phone number to call, display a phone call status screen, and get other information from the phone system. (See *6.3.1 THE PHONE STATEMENT*.)

Quota (QUO) statements increment quotas in the QUO file.

Special Information (SPC) statements get information from the operating system, the quota file, or the scratch areas, and do some other special functions.

System (SYS) statements let you temporarily leave Survent to talk to the outside world as well as some other special functions.

- Call other questionnaires

Call (CALL) statements allow each main questionnaire to have sub-questionnaires that can be called several times during the run.

3.1.1 Questionnaire Flow Control Statements

During an interview, it is often necessary to ask a set of questions *only* under certain conditions. There are three ways to do this:

1 You could restrict the asking of each question with the IF condition on the question. But, this would force you to type the same condition on all the questions of that section of the questionnaire.

2 You could skip a set of questions using the SKIPTO option on a CAT or FLD question response list. This is easy to do, but limits you to just one condition under which to move past the section.

3 You can use a GOTO statement in conjunction with an IF condition. This allows you to specify a complex condition and skip as large a section as you would like with just one statement.

GOTO Statement

- The syntax for a GOTO statement is:

GOTO,`label`

The label specified is the label of another question (FLD, GOTO, etc.) or compiler directive (ROTATE, END_ROTATE, END_GRID, or END_LOOP).

Here is a sample GOTO statement sending the interview forward to another question.

```
EX:
{CARDS:
!IF CARDTYPE(6) AND NUMITEMS(CARDTYPE) = 1
!GOTO,VISA }
```

If question CARDTYPE has a response of 6 and only one response, the program will go to question VISA.

GOTO statements can also be used to move backwards in the interview. RESET statements (see next section) are used to go back to a previously asked question, and automatically delete any intermediate data collected. Using a GOTO statement to move backward in a questionnaire leaves the original data intact, with the potential for dirty data and/or corrupted skip patterns. If you must use a GOTO statement to go backwards (and save old data), test the questionnaire thoroughly before interviewing. If your program causes you to go backwards more than 1000 times, it will abort with a BLOW error that will prevent you from being caught in an infinite loop.

GOTO statements are also used as a place to go to. Putting a GOTO statement at a specified place without specifying a label to go to will simply continue the interview at that place. This is often referred to as a receiving GOTO statement. Here is a sample receiving GOTO statement:

```
EX:
{
!IF items$="oranges"
GOTO,FRUIT}
.
.
.
{FRUIT:
!GOTO}
```

In the above example, FRUIT is the receiving GOTO. The receiving GOTO is useful when you need to send the interview to a particular place that you can't specifically address (i.e., send to the end of the questionnaire).

Text may be included on a GOTO statement. If it is included, the next question that displays must not clear the screen, or the text will be lost. Using the IF condition and the screen format controls such as boxes, you could paint the screen differently under various conditions, without having a different data entry question for each optional screen display. This function can also (and preferably) be accomplished with the DISPLAY statement subtype 2.

RESET Statement

RESET (or RSET) statements are used to return the interview to a previously asked question. This could be a question that caused the interviewer to enter a response, a data-generating question (GEN, EXP), or a labeled compiler directive. Data is removed from all questions asked between the RESET statement and the question reset to. This is generally used if some mistake in logic has been detected and you wish the interviewer to re-ask a set of questions. You can also use this to format your own error messages and re-ask a question in places where the standard error messages will not suffice.

The syntax for a RESET statement is:

!RESET,`label or -#`

RESET statements usually have an IF condition, i.e., the interview would only be returned to the specified question if the condition was true. Here is an example:

```
EX:
{NEXTQ:
!IF LAST$="NEVER" AND COST<=10
!RESET,FIRST }
```

If question LAST has a response of NEVER and question COST has a numeric response less than or equal to 10, the program will immediately return to question FIRST, removing previous responses between FIRST and NEXTQ.

A RESET statement usually includes some text with a message to the interviewer. If you include text, the interviewer will have to press **Enter** before the program resets. Here is a sample RESET statement using text:

```
EX:
{NEW:
!IF USE(2)
\(23)The respondent uses \:BRAND: brand, re-ask
questions
!RESET,NEWBRAND }
```

If question USE has a response of 2, the interviewer will see a screen with the above text. This text displays the brand name chosen. The program does not display a message to "press **Enter** to continue", so you might want to include instructions in the question text. The program will then return to question NEWBRAND, removing any responses to questions between NEW and NEWBRAND.

With RESET statements there are a few rules to keep in mind:

- The question that is returned to NEWBRAND or FIRST (in the previous examples) *must* have been asked previously.
- When it resets to a question, the program deletes the response to the returned to question and all subsequent responses, restoring the data to its state before this question was first asked.

- If you use a RESET statement to go forward, it acts like a GOTO statement.
- You may not use a RESET statement to move into or out of a SUSPEND, RESUME or SPECIAL block (See *3.2.1 INTERVIEW CONTROL COMMANDS*).

If, instead of specifying a label to reset to, you specify a dash followed by a number (from 1-27), Survent will go back that many answered questions.

```
EX:
!RESET, -5
```

This will reset back five questions (the last five answered/executed). This is very useful in a Terminate block, where if the interviewer really shouldn't have terminated, you can programmatically back them up to where they were when they mistakenly entered

3.1.2 Screen Control Statements

DISPLAY Statement

DISPLAY (or DISP) statements are used to display screens without expecting a data response. Possible functions include starting a section with interviewer instructions, conditionally painting different screens before a data entry question or placing a message on the screen for a set length of time. No data entry prompt (-->) is shown.

The syntax of the DISPLAY question is:

```
!DISPLAY,subtype,option
```

DISPLAY question subtypes are:

1 Displays the text on the screen and waits for **Enter** to be pressed to continue. This is the default subtype.

2 Displays the text and goes to the next question without waiting for any interviewer response. This is used to paint the screen with text before asking the next question, which typically prints its text below the DISPLAY text.

3 Displays the text on the screen, pauses for some seconds, and continues to the next question. You must specify the number of seconds to pause as the option. 60 seconds is the maximum. The seconds can be specified as whole numbers or decimal (1.5) numbers.

6 Like 3, but will allow the interviewer to press **Enter** to continue before the set number of seconds has passed.

For the standard DISPLAY subtype 1, the text screen comes up and after reading it, the interviewer presses **Enter** to continue. The interviewer can use one of the interviewer keywords at this screen (^, ABORT, RESET, RETAKE, SHOW, SUSPEND).

The DISPLAY subtype 2 gives you complete control of the interviewer screen when you find that you need to have additional text display or need to put text in different positions on the screen.

When using a DISPLAY,2, make sure that the next question asked does not clear the screen when displayed, or the display text will be overwritten by the next question's text. Here is an example using DISPLAY subtypes 1 and 2:

```
EX:
{
The following is a list of products. Please read
the name of each product to the respondent and
record which one they like best.
PRESS Enter TO CONTINUE
!DISPLAY,1}

{!ROTATE,R}
{
\A01 APPLES
!DISPLAY,2}
{
\A02 ORANGES
!DISPLAY,2}
{
\A03 BANANAS
!DISPLAY,2}
{!END_ROTATE}
\A99 Other
!DISPLAY,2}
{
\ADK Don't Know (Do Not Read)
!DISPLAY,2}
{
\A
!FLD,N
01 Apples
02 Oranges
03 Bananas
99 Other
DK Don't Know}
```

This set of statements would first display the interviewer note about rating the products, and wait for an **Enter**. Note that the instruction to press **Enter** was entered as part of the DISPLAY text; the question itself does not print any instructional text. Then it would display the three products in random order, and the interviewer would record the favorite. Notice that the rotation keeps the Other and Don't Know at the bottom.

DISPLAY subtype 3 is used to display a screen for a set time. You could, for instance, display a Thank you message for 10 seconds at the end of the interview. After the set time, the next question will appear automatically.

```
EX:
{
Thank you for your time...
!DISPLAY,3,10}
```

GRID Question Block

A grid question block is a set of questions that are presented on the screen at the same time to be filled in by the interviewer. It could be used any time a forms type screen is desired. Typical applications include:

- Filling in the respondent's name and address

- Having a set of numbers add to 100%
- Gathering name/age, etc. for various household members
- Providing data entry screens
- Filling a tax form
- Spreadsheet type applications

The grid block begins with {!GRID} and ends with {!END_GRID}. The {!GRID} question can include an IF condition and text; if the IF condition is not met, the entire grid is skipped. If it is met, the text on the {!GRID} question displays followed by the text on the other questions in the grid, and the interviewer is placed on the first data asking question after the {!GRID}.

While the user is on the grid screen, they can move from question to question by pressing **Enter** (either before or after a response) to go to the next question, **Tab** to go down, or by using the arrow keys (DOS, UNIX) or the **Ctrl** keys (UNIX, **Ctrl-U**, **D**, **R**, or **L** for up, down, right, or left). **HOME** and **END** are also supported (DOS/UNIX) and **Ctrl-G** or **Ctrl-E** (UNIX) to move to the top or bottom of the screen. When you get to the bottom of the screen, **Enter** will take you out of the grid if you have met all the grid requirements, otherwise it takes you to the top of the grid.

When the interviewer is done with the grid screen, pressing **ESC** will let them out of the grid, as long as any IF condition on the {!END_GRID} is satisfied. Otherwise, it will display any text on the !ENDGRID on the screen and then they are returned to the grid screen with the previous answers intact and must continue to change responses until the conditions on the {!END_GRID} are met.

If you back up into the grid or reset to the {!END_GRID}, the entire answered grid is redisplayed and you are set at the top upper corner of the grid, from where you can continue filling entries. If you back up out of the grid from there or reset to the {!GRID} question later, the entire grid is cleared. You cannot RESET into the middle of a grid. You cannot have a grid inside a rotate and you cannot have ROTATE statements inside a grid (except in WEBSurvent). However, you can have a !GRID statement in !RESUME, !SUSPEND, !SPECIAL and !AFTER_QUIT blocks.

If you execute a GRID on questions with existing data in their fields, the data will be filled in when the grid is displayed; in this way you may easily edit entries in your data file.

The caret (^) key inside of a grid will back you up to the question before the grid. You cannot use it to back up within the grid, use the arrow keys. The caret will back out of the grid from anywhere in the grid.

To refresh the screen in the grid, use **F3** (monitors) or **Ctrl-F-3** (terminals).

The syntax of a grid question is:

```
{label:
!IF condition
text
!GRID <,R or M> }
...
{
\(box spec)
text
!data asking questions in the grid
}
...
{label:
!IF condition

text
!ENDGRID}
```

The {!GRID} question itself is like other displayed questions, having an IF condition and standard box, screen display, and screen controls. The label can be used to programmatically skip to or reset to. The IF condition controls whether or not the entire grid is executed.

GRID,B allows some or all of the questions in the grid to not be answered (blank).

GRID,R re-executes all the grid questions after each response. This is used for “running total” applications. You may combine types **B** and **R**.

GRID,M says that you must answer all questions in the grid according to their specifications before you can **ESC** out. This is the default.

GRID,E,<grid label> lets you edit a previous grid. It places you back in the grid with all previous answers intact. Modify as you wish. **ESC** exits the grid, depending on the grid answering requirements. Do not use an ENDGRID with a GRID,E.

Standard Open-End Grids

In standard Survent, the questions inside of the grid will each have a box specification to position them in a certain place on the screen (in WEBSurvent this is not required. The questions are listed from the top down). Most questions will also use the _ option to position where the cursor will be for a response.

```
EX:
{
*** ENTER RESPONDENT INFO HERE. PRESS <ESC> WHEN
YOU ARE DONE.
*** YOU MUST ENTER AT LEAST THE NAME, ADDRESS, AND
CITY/STATE.
!GRID,B}
{R:
\(5,,5)
Respondent name: \B\______...............\E
!VAR,,20,5}
{A1:
\(7,,7)
Address 1 : \______...............
!VAR,,20,5}
{A2:
\(9,,9)
Address 2 : \_....................
!VAR,,20,0}
{C:
\(11,,11)
City/State : \______...............
!VAR,,20,5}
{Z:
\(13,,13)
Zip code : \______....
!VAR,N,9,5}
{!IF R$<>" " AND A1$<>" " AND C$<>" "
Must enter Name, Address, and City at the very
least!
!END_GRID}
```

This grid shows a few of the intricacies of screen building. The {!GRID,B} question includes some initial text to put on the screen. The "B" allows some questions to be blank.

The questions are each specifically placed on a separate line; notice that you must enter the starting line and the ending line, otherwise the question will clear the rest of the screen when you get to it in the grid. Each question uses _ to position where the cursor will go, followed by a series of dots and dashes to identify how many characters are required and how many are allowed maximum.

Notice the \B on the first question prompt. This means bold, and will cause both the prompt initially and then the typed text to be bold. Another frequently used highlight is the \I for inverse, which makes a nice data entry box to type into.

The interviewer may move around and fill out the entries as they choose; when they press **ESC** they will get the message about entering the Name, Address, etc. unless they have filled those values in.

Initial Screen Setup Using a Grid

Another technique used for building grids is to paint the entire screen on the {!GRID} question, and position the cursor within the grid:

```
EX:
{
INTV: FILL ALL ENTRIES FOR EACH CHILD MENTIONED:
Sex Age Education
(M,F,D) (0-60,RF) (0-25,K,RF)
First child:
Second child:
Third child:
```

```
Fourth child:
!GRID}
>REPEAT $A=6,8,10,12
{
\($A,18,$A,22)\I\_ \E
!CAT,N
F Female
M Male
D Refused
}
{
\($A,34,$A,38)\I\_ \E
!NUM,Z,,0-60,,RF}
{
\($A,50,$A,54)\I\_ \E
!NUM,Z,,1-25,,K,RF}
>ENDREP
{Endgrd: !ENDGRID}
```

In this setup, the cursor is positioned on a different line for each child and the answers are written in an inverse video box.

Totaling Numbers in a Grid

Something often done with grids is to make sure that the numbers on the screen add up correctly before moving on. Here is an example of how to do a grid that sums numbers to 100:

```
EX:
{
For each grocery store used, enter the percent of
the time you use it, or 'DK' if you are not sure.
The total should add to 100%.
Store 1: ___ %
Store 2: ___ %
Store 3: ___ %
Store 4: ___ %
Total: ___ %
Left: ___ %
!GRID,R }
{ STORE1:
\(5,11,5,14)\_\U \E
!NUM,,,0-100,,DK }
{ STORE2:
\(7,11,7,14)\_\U \E
!NUM,,,0-100,,DK }
{ STORE3:
\(9,11,9,14)\_\U \E
!NUM,,,0-100,,DK }
{ STORE4:
\(11,11,11,14)\_\U \E
!NUM,,,0-100,,DK }
{ TOTALX: .3
!EXPR,,X(STORE1)+X(STORE2)+X(STORE3)+X(STORE4) }
{ LEFT: .3
!EXPR,,100-TOTALX }
{
\(13,11,13,14)\U\:TOTALX:\E
!DISPLAY,2 }
{
\(15,11,15,14)\U\:LEFT:\E
!DISPLAY,2 }
{ BADSUM1:
!IF TOTALX > 100
\(17,11,18,79) SUM OF ANSWERS = \:TOTALX: IS
GREATER THAN 100
!DISPLAY,2 }
{ BADSUM2:
!IF TOTALX<100 AND NOT([STORE1$"DK "," "] OR &
[STORE2$"DK "," "] OR [STORE3$"DK "," "] OR &
[STORE4$"DK "," "])
\(17,11,18,79) SUM OF ANSWERS = \:TOTALX: IS LESS
THAN 100
!DISPLAY,2 }
{
!IF TOTALX=100 OR (TOTALX<100 AND &
([STORE1$"DK "]OR [STORE2$"DK "] &
OR [STORE3$"DK "] OR [STORE4$"DK "]))
!ENDGRID }
```

In this example, the user would enter numbers until they had reached 100% or at least responded "Don't know" to some store percent. The running total is redisplayed after each question because of the use of the {!GRID,R} to repaint the screen.

Spreadsheet Applications, Complex Grid

A spreadsheet-type application would have text around the sides of the grid, with the middle of the grid reserved for cursor movement and typing entries. In this case, there are a number of additional features that you should use. Here is the extended syntax:

```
{
text, \_
!GRID, [R], [#A], [#B], [#C], [R/C], ([#R],[#C],#GR,#GC)
```

Where:

#A = number of questions in each column,

#B = number of questions in each row,

#C = spaces for each question

R/C (Rows first/Columns first)means after pressing <Return> at a question, go to the next question in that row if R or column if C.

([#R],[#C],#GR,#GC) is a box spec for the questions in the grid, which otherwise defaults to the rest of the screen to the right and below the _ on the grid text, otherwise the rest of the screen below the grid text.

When #A and #B are specified, each question in the grid is expected to fit into a box:

```
(#qs per column/total columns )by(#qs per row/total rows)
```

For example if you had 10 questions and you said there were 5 columns by 2 rows, using a standard 24-inch-by-80-inch screen, each question would get a box size 16 columns (80/5) by 12 rows (24/2).

In addition, if a question inside the grid has a box specification, its specification is relative to the '_' in the grid text, not relative to the whole screen.

The maximum number of items in a spreadsheet-type grid is 48; e.g., 4 by 12 or 5 by 9 or 6 by 8. Each row must be separated by one blank row.

Here is an example spreadsheet:

```
      {
      \T
      ENTER PERCENT OF TIME SPENT ON EACH ACTIVITY. EACH ROW
      MUST ADD TO 100.

            Eat   Work  Read  Exercise Nothing TOTAL
            ---   ----  ----  -------- ------- -----
1) \:Name1^9 \_
2) \:Name2^9
3) \:Name3^9
!GRID,R,3,5,10,R,(7,16,13,74) }
>REPEAT $A=01,...,15
{ v$A:
!NUM,,,0-100 }
>END_REPEAT
{ Row1tot:
!EXPR,,X(V01)+X(V02)+X(V03)+X(V04)+X(V05) }
{ Row2tot:
!EXPR,,X(V06)+X(V07)+X(V08)+X(V09)+X(V10) }
{ Row3tot:
!EXPR,,X(V11)+X(V12)+X(V13)+X(V14)+X(V15) }
{
\(8,66,9,70)
\:Row1tot:^-3
!DISPLAY,2 }
{
\(10,66,11,70)
\:Row2tot:^-3
!DISPLAY,2 }
{
\(12,66,13,70)
\:Row3tot:^-3
!DISPLAY,2 }
{!IF Row1tot = 100 AND Row2tot = 100 AND Row3tot = 100
!ENDGRID }
```

In this example, the names of the people filled in on the grid screen are displayed in a field nine wide (\:Name1^9), and the totals are displayed right-justified in a field three wide (\:Row1tot^-3).

The user will move from question to question, across the screen as entries are filled in ('R' in !GRID controls). Because {!GRID,R} is used, the screen will be updated each time a new entry is typed. Once all the rows add up to 100 percent, you can press **ESC** and move on to the next question.

Note: Screen refreshing can be CPU intensive if you are working in supervised Survent. Please test this before committing to using a {!GRID,R} in a live interviewing situation.

3.1.3 Data Generating Statements

EXPRESSION Statement

Expression, or EXP, statements let you specify an arithmetic expression or a string value. For arithmetic, the program calculates it and places the result in the data location for this question.

Expression questions will usually be labeled since you will likely want to see or check the value you have recorded later during the interview.

You should specify a length on EXP statements. Otherwise the system will use the default width of 9. Specifying the header option NUMERIC_WIDTH_REQUIRED will require that you specify a length.

For numeric expressions, if the result does not fit in the field, the field will be filled with asterisks (*). The maximum width is 20. The format includes up to a 19-digit integer with or without a plus (+) or minus (-) sign, an optional decimal (.), and up to eight decimals of significance. By default, the !EXP statement will look at the expression and determine the maximum number of decimals in any constant or NUM question referenced and use that as the number of decimals to record.

!EXP questions will accept a “number of decimals to save” parameter option similar to !NUM questions. The syntax for an EXP statement is:

```
!EXP,subtype,numdecs,expression
```

The subtype is optional; however, if it is not specified, a comma must be used as a placeholder. A comma placeholder is not required for numdecs.

The EXP statement subtypes are:

B Blank-filled, right-justified numbers. This is the default. (e.g., " 12")

S Writes a "string" to data instead of a number

Z Zero-filled, right-justified numbers. (e.g., "0000012")

Numdecs lets you specify the number of decimals the answer will be stored with. It can be a number from 1 to 8, or "Nodecimals". If it is 1-8 it stores the value with a decimal point in the data. If you specify NODECIMALS, the answer is stored as a rounded whole number, even if the expression has values with decimals in them.

The expression may include any combination of the following:

- Numbers
- Arithmetic operators (see below)
- References to previous questions or data locations
- Functions (see *2.6.4 USING FUNCTIONS IN CONDITION STATEMENTS*)
- Any number of parentheses (to control the order of execution of the expression)
- A "quoted string" to save in the data if the subtype is "S"

The arithmetic operators and their order of calculation are:

Operator	Type	Order of Calculation
**	exponentiation	1
*	multiplication	2
/	division	2
+	addition	3
-	subtraction	3

The order of calculation means that if several operators are used in the same expression without parentheses to prioritize the order, the operator's type will determine the order. If using two operators with the same order level, the expression is calculated left to right.

Calculations can be done in real arithmetic as well as decimals. The numdecs parameter determines how much decimal significance is kept. If any part of the equation is missing, the result will be filled with asterisks.

Here are some other sample expression statements:

```
EX:
{TOTMEALS: .4
!EXP,,DAYS * MEALSPER}
```

This statement will multiply the response to question DAYS by the response to question MEALSPER. The saved data will use four columns (.4). No subtype is specified, so the result will be right-justified with leading blanks.

Expressions are often used to update counters:

```
EX:
{[TOTCARS]
!IF NUMFORDS > 0
!EXP,Z,TOTCARS+NUMFORDS}
```

This statement adds the response to NUMFORDS to TOTCARS if NUMFORDS had a numeric response greater than zero. The Z subtype stores the data zero-filled. This statement has additional special features. First, it uses the data location previously used by the question TOTCARS ([TOTCARS] as the location). This allows you to use the same data space over and over, instead of using a new space for each EXP question updating TOTCARS. \|TOTCARS backreferences will show the updated number from this point on.

```
EX:
{!IF ADDROOM(YES)
!EXP,,NUMITEMS(ROOMS)+1}
```

This statement will count the number of responses to the CAT or FLD question ROOMS and add 1 to it if the answer to ADDROOM is YES.

```
EX:
{TIMES:
!EXP,,MAX(WEEK1,WEEK2,WEEK3,5)}
```

This statement will find the highest numeric response among the questions WEEK1, WEEK2, WEEK3, and the number 5 and return its value.

```
EX:
{FAMILY: .3
!EXP,,X(CHILD) + X(ADULT)}
```

This statement will add the responses to questions CHILD and ADULT. If either question is blank or not numeric, the numbers will still be added because the X function causes a missing response to return a zero. Otherwise, if either question were blank, no response would be recorded for this calculation. The answer will be stored in a three-column-wide field.

```
EX:
{CARNAME: .20
!EXP,S,,"Chevy"}
```

This statement will save the name “Chevy” into the data. It will also save backreferences to other questions like “\:Carshave”.

GENERATE Statement

Data can also be generated by the program with a Generate, or GEN statement (also see SPC,9; *3.1.5 SYSTEM INFORMATION STATEMENTS, Special Information Statements* for more information). Data can be moved or blanked. CAT and FLD question responses can be added, removed or netted. GEN statements, such as GOTO and EXP statements, are frequently based on condition statements.

The general syntax for a GEN statement is:

```
!GEN,subtype,label or location,response code or
punch or length,offset
```

GEN statements require a subtype and a receiving question. The rest of the parameters depend on the subtype. If a data location reference is used, brackets ([ ]) should be placed around the data location.

Note that back-references to CAT or FLD questions previously asked will still show the response at that time, even if the question's data has been altered or blanked with a GEN statement. Place a CAT or FLD question subtype of A with an ALIAS= parameter on the updated location to update the answer array.

All GEN subtypes have a maximum of 3000 columns that can be accessed at one time. If you need more than that, simply do multiple GENs.

The GEN statement subtypes for general use are:

B Blanks question(s) or data

M Moves (copies) question(s) or data

The subtype for TEX questions is:

U Blanks the text pointer and clears the text from the TEX question.

The subtypes for CAT and FLD questions are:

A Adds a response to a response list.

Z Zaps (removes) a response from a response list.

O Nets responses from one question to another.

T Takes responses out of one question that were in another.

Q Replaces code in a question with the code(s) in other question(s).

R Replaces code in a question with some other code(s).

Other subtypes are:

C Compresses data.

E Erases the answer array.

GENERATE Subtypes for General Use

GEN statement subtype B is used to blank data. Use caution with this because the data cannot be recovered once blanked. This is used to blank temporary data from a section of a data file or when going backwards using GOTO statements.

The syntax for a GEN,B statement is:

```
!GEN,B,label or location,length
```

If you reference a question label, the default length is the question’s length. The default length for data locations is one.

```
EX:
{!IF NUMHOUSE < 2
!GEN,B,[2/1],80}
```

This would blank the data starting in record 2, column 1, for a length of 80 if the response to NUMHOUSE was less than 2.

GEN,B cannot be used on a TEX type question. Use GEN subtype U to blank TEX questions.

GEN subtype M copies data from one question or location to another. Any data already in the receiving location is replaced with the data from the sending question. GEN subtype M may also be used to move TEXT data to other TEX questions or to and from VAR questions.

NOTE: When writing from TEX to VAR questions, use the offset feature to access columns beyond 3000. The maximum size of any question is 3000.

The syntax for a GEN,M statement is:

!GEN,M,to label or location,from label or location,length

or, for writing long TEX questions to VAR questions:

!GEN,M,to label or location,from TEX question label,length,offset

GEN,M replaces any data that is already in the question it moves a response to. The sending field still has a copy of the data after the move has taken place. The length is optional. It defaults to the length of the receiving question if using labels, or one if using data locations.

```
EX:
{!IF X(FEB) < X(JAN)
!GEN,M,NEWBRAND,BRAND }
```

If the response to question FEB is less than the response to question JAN, the program will move the data from question BRAND to question NEWBRAND.

If you have a series of questions to move, using a length will let you move them in one block as opposed to moving the questions one by one.

```
EX:
{!GEN,M,NEWBRAND,BRAND,10 }
```

This statement copies all data starting at question BRAND for a length of 10 columns, to NEWBRAND for a length of 10 columns.

```
EX:
{!GEN,M,[3/23],[4/35],10 }
```

This says to move to data record 3, columns 23-32 (length of 10) what was in data record 4, columns 35-44.

```
EX:
{!GEN,M,[10.5],[23.5]}
```

This shows how you can specify the length of the move within the data location itself.

```
EX:
{!GEN,M,FIRST,[20],5 }
```

This would move columns 20-24 to the columns starting at FIRST, for a length of 5.

You can use the GEN subtype M to copy a TEX question to a VAR question or to another TEX question. To do this you must always reference the TEX question by its label.

You can copy a VAR question to a TEX question, but you must specify the length (GEN,M, tex, var, 76). To get all the data from a TEX question (which may have up to 5000 columns) to the regular data area, you must use “offsets” to override the maximum of 3000 columns per GEN statement:

```
EX:
{!GEN,M,[1001],TEX1,3000 }
{!GEN,M,[4001],TEX1,2000,3001 }
```

This would move the 5000 columns of data from the TEX question TEX1 to locations starting at 1001. Notice the “offset” of 3001 on the second GEN statement to get data starting in the 3001st position from the question TEX1.

You can also write from !TEX to a !VAR with the following:

```
{var1: .3000 hide
!var}
{tex1:
!text}
{!gen,m,var1,tex1}
```

But, if you have more text than a !var can hold (3000 columns), you need to grab pieces of text (3000 columns at a time) with the following:

```
{var1: .3000 hide !var}
{var2: .2000 hide !var}
{tex1:
!text}
{gen,m,var1,tex1,3000}
{gen,m,var2,tex1,2000,3001}
```

GENERATE Subtype for TEX Questions

GEN subtype U is used to remove text pointers from the data, the associated text in the text data area, and the answer from the answer array, leaving the space available in the text data area.

The syntax for the GEN,U statement is:

!GEN,U,<u>label or location</u>

Since text pointers always have a length of one, no length is specified. If you try to use GEN,B on a TEX question, you will get a compile error in PREPARE. If you blank a TEX question with a GEN,U statement you may RESET over it, but the text blanked by the GEN is not restored.

GENERATE Subtypes for CAT and FLD Question Responses or Data Column Punches

GEN subtypes A, Z, O, and T and R and Q are used to act on the responses to a CAT or FLD question’s response list or the punches in specific data columns. Subtypes A and Z use specific response codes or punches. Subtypes O and T net together or take out responses or punches from a question or location depending on the contents of another question or location. Subtypes R and Q replace codes in a code list with other codes or codes from other questions.

The GEN statements used with CAT or FLD questions (A, Z, O and T) are particularly useful to present multiple questions using the same question list, but requiring different responses depending on a prior response. Examples of this are:

1 First response, then subsequent responses to a list

2 picking the favorite out of a list of previously chosen items

3 Displaying new text based on a prior response

These all use GEN statements in combination with CAT subtypes A, C, and D (or subtype I without GENs). Examples of these uses can be seen in *Appendix Z: PRACTICAL APPLICATIONS*.

The syntax for GEN statement subtypes A and Z is:

!GEN,A or **Z**,<u>label or location</u>,<u>codes or punches</u>

If a label is used, a response code that exists in the question's response list is required. If a location specification is used, a punch (1-9, 0, X, or Y) is required. See *2.6.2 USING DATA LOCATION REFERENCES* for a discussion of punch references.

When adding or removing (zapping) responses or punches, other responses or punches are not affected.

```
EX:
{!IF OLDBRAND(1)
!GEN,A,BRANDS,3 }
```

If question OLDBRAND had a response of 1, the program will add a 3 response to question BRANDS.

```
EX:
{!GEN,Z,BRANDS,1 }
```

This statement removes a 1 response from question BRANDS. Any other responses will remain in the data.

```
EX:
{!IF SEX(M)
!GEN,A,[2/20],8 }
```

This adds an 8 punch to record 2, column 20 if SEX was answered with M. Other punches in that column would not be affected.

For multiple-response FLD questions, the A GEN will add the new response into any remaining available blank columns of that FLD question. If there is no more room, Survent will abort with a blow error. The Z GEN will remove the response then left-justify the remaining responses in the FLD question's data location. The M writes over everything that was there, while the O nets responses.

GEN O (Or net) and T (Take out) statements work like subtypes A and Z, except that rather than referencing specific codes, data is netted or removed based on the data in another question.

The syntax for GEN subtypes O and T is:

> **!GEN,O** or **T**,to label or location,from label or location,-length

The receiving question or location and sending question or location are required. The length is used if you have more than one contiguous question or set of columns to act on or you only want to affect part of a question. When specifying a length, each column of the sending question or location is netted or removed from its corresponding column of the sending question or location.

The default width, if none is specified, is the width of the labeled question, or one if using data locations.

GEN subtype O nets the responses from one sending question to another. To net together multiple questions, simply add more GEN,O statements. For instance, one might wish to net the response to the "best" pizza question to the "other" pizzas known about question, to see all pizzas known about.

If a dash (-) is placed before the length on a GEN subtype O, all the columns of the sending question and length will be netted into the first column of the receiving question. This is useful for one column wide CAT question response lists or data locations.

```
EX:
{
!GEN,O,NEWBRAND,BRAND }
```

This statement nets the punches from the question BRAND to NEWBRAND. After the net, NEWBRAND will have its original responses as well as any different responses from question BRAND.

```
EX:
{
!GEN,O,20,18,2}
```

This statement adds the punches in columns 18 to column 20 and the punches in column 19 to the punches in column 21.

```
EX:
{
!GEN,O,39,23,-3}
```

This statement nets the punches from column 23-25 (23,-3) all into column 39 (because of the -).

For multiple-response FLD questions, the O GEN will net the new responses into any remaining available blank columns of that FLD question. If there is no more room, Survent will abort with a blow error.

GEN subtype T takes all the responses out of the receiving question that are in the sending question (or all of the punches if referring to locations). Only those responses unique to the receiving question will remain. This is the opposite of the GEN,O netting function.

```
EX:
{
!GEN,T,NEWBRAND,BRAND }
```

This statement takes all responses out of question NEWBRAND that were in question BRAND.

Only responses unique to NEWBRAND will remain.

For multiple-response FLD questions, the T GEN will take out the duplicate or matching responses then automatically left-justify the remaining responses in the FLD question’s data location.

GEN,Q and GEN,R subtypes allow you to replace a code in a code list with other codes from other questions. These can be very useful in coding situations, where you want the new code(s) to replace the “OTHER” code in the question being coded.

The GEN,Q statement replaces a code in a question with codes entered in a different question.

The syntax is:

```
!GEN,Q,question changed,code replaced,questions
from

EX:
{!GEN,Q,Q23,95,Q23otcd1,Q23otcd2}
```

This statement would replace the 95 code in Q23 with the codes entered in Q23otcd1 and Q23otcd2.

You can have as many codes as you want in the “from” questions, it will insert those codes in the list at the point you are replacing the code(s).

The other statement similar to a !GEN,Q is a !GEN,R. The !GEN,R statement replaces a specific code in a question with a different code or codes. The syntax is:

```
!GEN,R,question changed,code replaced,new code(s)

EX:
{!GEN,R,Q32,95,97,98}
```

This statement would replace code 95 with codes 97 and 98.

Both of the !GEN,Q and !GEN,R keep response order intact on FIELD questions. You can add any code that is the proper width for the question, even if it is not on the code list for the receiving question. If you try more codes than will fit in the question space, it just adds as many as it can.

Miscellaneous GENERATE Subtypes

If you should need to blank out the answer array for a question, you can use the GEN subtype E. This will not affect the data, just what shows on the screen when you do back-references.

The syntax for a GEN,E statement is:

```
!GEN,E,label
```

Use this to eliminate 'Other' responses when doing ALIAS=xx, for instance. A GEN,C can be used to compress data.

The syntax for a GEN,C statement is:

```
!GEN,C,label/location.width of codes, width to
compress
```

This could be useful if you used separate adjacent questions to collect answers and you now want to move the data so it is in one field with no blanks (i.e., “01 02 05” becomes “010205”). This could be handy to set up codes for a FLD,I reference later on in the questionnaire.

Suppose you had a string of two-digit codes in a 10-column field and you wanted to remove any embedded blank fields from the string. The code would look like:

```
{Colors: .10 Hide
!fld,,5
01 Black
02 Blue
03 Brown
04 Cyan
05 Green
06 Red
07 Blue
08 White
09 Yellow }
{!Gen,C,[Colors.2],10 }
```

This GEN type is intended for use only to collapse codes across multiple field questions or with a data field where all questions and gaps are coded with the same length. Do not use this to compress variables of different code or value lengths!

The program uses the starting location and length to determine whether to collapse the following fields. If a following field is blank for the duration of the first question's width, it is compressed out, otherwise it stays in its relative position.

The GEN,C has no column limit. You can compress as many codes as you want. However if you compress more than 240 columns worth of codes and then you back up over the GEN,C, Survent will not remove the codes.

3.1.4 Data Entry Question Function Modifying Statements

Disk-Based Field Questions for Large Response Lists

Disk-based Field (or FLD) questions allow you to have up to 32,000 categories (the standard FLD question only allows around 800). They are called disk-based because the question is compiled separately from the questionnaire and the response list is stored as a separate file on the disk.

The syntax for a FLD question using a disk-based response list is:

```
!FLD,subtype,,filename
```

Only subtype A is allowed; N and R are ignored. The question must be single-response. This question type does not work with HIGHLIGHTCATS mode.

If no subtype is specified, the question is displayed on the screen by level, and will return a final code after all of the levels have been asked. You can use this with a FLD subtype A to associate text with long lists of product codes, quota subgroups, etc.

The extra comma after the subtype is required. It is a placeholder for the number of responses.

To do a similar function with multiple responses (multiple levels), see the O= response list option.

The file name can be any valid file name (starting with a letter), with or without an extension. By default it uses an extension of “dbr”. Use “$filename” to refer to a file with a different or no extension.

An example of a use for this would be when asking about a product with model numbers within car brands. The part of the list referring to the brand would be displayed first. Then, the model number sub-level would appear. The code stored would be a unique code that identifies the brand AND the model number.

During interviewing it may be necessary to change a long list, for instance, to accommodate unexpected mentions that are falling into “miscellaneous.” With a disk-based FLD question, you can change the list while interviewing is going on without having to re-compile the whole questionnaire.

Designing and Compiling the Disk-Based Response List File

You compile a disk-based FLD question response list with the command:

```
~PREPARE DBR
```

You can also spell DBR out as DISK_BASED_RESPONSE_LIST. The next statement after this would be "OUTPUT=filename", specifying the name of the disk file that will hold the response list.

This must be the same name that you have on the !FLD question statement in the questionnaire. You may have as many references to the same file as you would like in the questionnaire, and can have as many different disk-based response lists as you would like. If you have more than 1000 expected terminal codes in the DBR file, you must specify the number of entries by saying "OUTPUT=filename, ENTRIES=#" where # is the number of codes. Although we have used the extension DBR in the following example, the extension is optional and, if specified, can be any valid name. The equal (=) sign on the disk-based recode table OUTPUT statement is optional; a space will also work.

This file is written with levels. An item in the list may have other levels beneath it. The terminal item of a path must have a terminal code to be put into the data. Terminal codes must be the same length and may be up to 20 characters long. There is a limit of 10 levels. Each level must fit on the screen if it is presented; this restricts the number of items allowed on a level.

When going live, the .dbr file needs to go into the $CFMCQFL directory (usually $CFMC/qff/).

Each level is lined up starting in the same column. A new level must be indented at least one space in from the previous level. There is special syntax to print text for a level, and other controls.

Here is an example of a disk-based response list:

```
EX:
~PREPARE DBR
OUTPUT=COMP1.DBR
      01 IBM
001 (SKIPTO QQ4) A 360
            B 370 \BTESTING\E
002   X- 1
003   Y- 2
004   Z- 3
005   C 1130
      02 HEWLETT PACKARD
            AA 3000
006   1 II/III QQ5
007   2 3X
008   3 4X
      4-6X
009         A 64
010         B 68
            BB 900 Series
011   1 925
012   2 935
013   3 Other
014   03 PERTEC
      04-MISC
015         AAA MODEL 1
016         BBB MODEL 2 QQ6
??? Unmatched code in data
END
~END
```

The first level of this response list is 01 IBM, 02 HEWLETT PACKARD, 03 PERTEC, 04 MISC, and ??? (a no match item). Code 03 is a terminal branch, 04 has one sub-branch, 01 has two subbranches, and 02 has three sub-branches. Also, the response code items may contain back-references to previous questions, screen enhancements, and skips to other questions. The terminal codes are listed starting in column one.

You will often need to put text on the levels of the response list. This is done by putting an exclamation point (!) and the text above the first response code of a level. There may be up to 20 lines of text per level (after the initial text for the question). You can use any of the text control statements to position the text, start a new line, highlight words, etc.

When the response to the question is later displayed, you will see the text from all of the levels responded to. You can keep a level's text out of the back-reference by putting a dash (-) in the space after the response code, as in X, Y, Z, 4 and 04 above. Answer text is concatenated for each level if necessary. The maximum answer text may not exceed 159 minus the number of levels. If it is longer, the error "no space for answer text" will print in Survent.

The ??? syntax as a terminal code allows responses in the data that don't match the other provided codes. This is so that data that has been pre-coded can be allowed into the questionnaire without rigid checks, which would be very difficult on a large list.

The END statement must be the last statement in the file before the ~END.

The FLD question that uses this response list looks like this:

```
EX:
{Comptype: .3
What is the computer you use most often?
!FLD,,,COMP1.DBR}
```

You must specify a data width; since the response list is compiled separately, PREPARE has no way of knowing how many columns to allocate unless you tell it. Neither PREPARE nor Survent will complain about a bad width. PREPARE will give an error if no width is specified.

The text from the question would display first followed by the text for the first level and its response items and so on.

When Survent is run on the above example, the question would first display level one items (01, 02, 03, 04). Typing 04 would display a second response list consisting of AAA MODEL 1 and BBB MODEL 2. Choosing BBB at this point would put 016 into the data and send the interview to question number 6 (QQ6).

Here is another example with text on each of the two levels:

```
EX:
~PREPARE DBR
OUTPUT=COMP1.DBR
      !WHICH OF THESE COMPUTERS DO YOU HAVE?
      01    IBM
      !IS IT A 360 OR 370?
001   A 360
002   B 370
      02    HEWLETT PACKARD
            !1, 2, OR 3?
003   X- 1
004   Y- 2
005   Z- 3
END
~END
```

Some final notes: When interviewing, you can use the caret (^) to walk back up a path to possibly go down another one, or even to get out entirely to the question before this one. You may also use the other interviewer keywords. See *4.3 Survent COMMANDS.*

When doing networked interviewing, the DBR file belongs where the QFF file goes.

There is no way to continue text in a disk based response list by using &, but the spec file can have up to 5000 characters per line.

SKIPTO may be specified preceding the text but indented for that level.

```
EX:
!This text will print at level A
A     LEVEL1 A
      (box spec) text and screen position for items
      at this level
      0 LEVEL2 Under A
A0XX              XX- LOWEST ANSWER
A0YY              (SKIPTO label)YY HIGHEST ANSWER
```

Note that the SKIPTO is indented correctly for this level.

To update (i.e., compile a disk-based FLD question with reference checking) an already compiled QFF file with your disk-based response list:

```
~QFF_FILE <studyname>
~PREPARE DBR
&specfile (or have actual specs here)
~END
```

NOTE: You do not have to recompile the specs for your main questionnaire, and you can update the DBR when the QFF file is being accessed.

You can also put your DBR compiles at the end of your regular compile.

NOTE: You do not have to recompile the specs for your main questionnaire, and you can update the DBR when the QFF file is being accessed.

```
EX:
~PREPARE COMPILE
[MAIN}
… questionnaire specs …
~PREPARE DBR
… DBR specs1 …
~PREPARE DBR
… DBR specs2 …
~END
```

On conditional statements, you cannot have references to DBR code lists, such as DBR1(001-004); they are too long for the program to store in memory. Instead, use [#] or [$] references; for exqample:

```
!IF [DBR1#001-004] or
!IF [DBR1$”AAA” - “AAZ”]
```

Pre-loading Responses in Disk-Based Recode Lists

Pre-loaded entry strings may be used; this means that you will be able to load (part of) a DBR response, and be set up in the DBR at the point where you have pre-set to. This is most useful if you have a sample file that may have some initial product information and may not, such that you could skip people through the initial screens if you had the data for that already.

The entry string can include direct response codes or back-references to the data of other questions. The syntax is:

```
!FLD,,,<name>.DBR,<\|label|stuff\|label2|junk,\|label3,
etc.>
```

Commas are discarded and are acceptable separators; spaces are not. Each part separated by commas will be treated as an answer to one screen of the DBR. You must back-reference the data of the prior question (using\|label|), because that is what is being accepted as the response, not the text of the response (\:label:).

EDIT Statement

You may need to allow an interviewer to edit a previously asked TEX or VAR question. For instance, you may want to review all previous open-ends after the interview is complete and allow the interviewer to 'clean up' the responses. The EDIT statement does this.

The syntax for an EDIT statement is:

```
     !EDIT,question label to edit
```

EDIT statements are frequently based on a condition statement—the interviewer would only be allowed to edit the specified question if the condition is met.

An EDIT statement may include some message to the interviewer, although it's not necessary. The interviewer will be placed on the response of the question asked using the full-screen editor (monitors) or line editor (terminals) to edit the question's response. See *4.2.1 RESPONDING TO DIFFERENT QUESTION TYPES* and *4.2.2 RESPONDING TO SPECIAL QUESTION TYPES* for details on editing text.

NOTE: Use the !GRID,E statement to edit GRIDs of questions.

Here are some sample EDIT statements:

```
     EX:
     {
     !IF FINISH(2)
     !EDIT,COMMENTS }
```

If question FINISH has a response of 2, the interviewer will be given the opportunity to edit question COMMENTS.

```
EX: {
!IF CHECKTEXT(COMMENTS)>0
!EDIT,COMMENTS }
```

This IF statement counts the number of characters in the text question COMMENTS. If the number of characters is greater than zero, the COMMENTS question is presented to be edited.

See *2.6.4 USING FUNCTIONS IN CONDITION STATEMENTS, Response Counting Functions* for more information on the CHECKTEXT function.

With EDIT statements there are a few rules to keep in mind:

• The question that is to be edited must have been previously asked.

• You may not use an EDIT statement in a GRID block.

• You may only edit TEX and VAR questions. *However,* TEX questions will only be edited if the have *not been* skipped. Otherwise, the !EDIT statement will not be executed and it will skip to the next question.

• If you back-reference the question after it has been edited, you will see the edited (not the original) text.

LLOOP/END_LOOP Statements

When asking multiple-response (CAT or FLD) or numeric (NUM, EXP) questions, you may need to repeat a question (or series of questions) for each answer (or number). You can accomplish this with a LOOP block that begins with a LOOP statement and ends with an END_LOOP statement.

The syntax for a LOOP statement is:

!LOOP`,control question label,maximum times through loop,amount of data`

pace per loop iteration You must specify the maximum numtimes through the loop; this can be up to 25. If you don’t specify the amount of data space to use, the program will do it for you. When using the header option CARD_FORMAT, you cannot have more than 75 characters of response in a loop because loops will not put data across case or card ID fields.

The syntax for an END_LOOP statement is:

```
!END_LOOP
```

Each response to the question specified as the control question label will trigger the series of questions in the loop. The maximum number of questions in a loop is 100. The control question must be a CAT, FLD, NUM or EXP. The data is generated in the order of the loop: first, the controlling response code or number is recorded, followed by the data from that loop iteration, then the next code, and so on. When using a multiple-response question as a controller, the loop questions are asked in the same order as the responses to the question were recorded, *not* in the order of the response list (unless in Highlightcats mode). An additional feature of loop blocks can be used when you are asking questions in a loop: any back-reference to the loop controller will reflect the response that is controlling the current loop iteration.

Here is a sample LOOP/END_LOOP block, with a CAT question as the controller:

```
EX:
{PHONES:
Which of the following types of telephones do you
have?
!CAT,,5
1 Wall phone
2 Desk phone
3 Specialty phone (Trimline, designer, etc.)
4 Novelty phone (Garfield, Mickey Mouse, Snoopy,
lips, etc.)
5 Cordless phone
(SKIPTO FONELESS) 6 None of above }
{
!LOOP,PHONES,5}
{PURCHASE:
Where did you purchase your \:PHONES:?
!CAT
1 From an AT&T phone store
2 From a department store
3 From an audio/video/electronics specialty store
4 From a mail order house
5 None of above }
{
!END_LOOP}
```

For each response to question PHONES (the control question), question PURCHASE will be executed once, up to a maximum of five times (if there were only two responses to question PHONES, the loop will be executed twice). Question PURCHASE also demonstrates backreferencing of a response; in this case, inserting the type of telephone in the text of question PURCHASE. The first response to question PHONES will appear the first time question PURCHASE is executed, the second one the next time, etc. in the order they were entered. Notice that the END_LOOP statement is required.

The code for the response being asked about in each iteration of the loop will also be stored in the data, in the order of response code, data, response 2, data 2, etc.

NOTE:

- Responses collected within a LOOP/END_LOOP block should not be back-referenced outside the block. If you do this, only the first response will appear. If you wish to back-reference these, you must place additional separate questions on the data (i.e., CAT,A).
- You cannot skip into the middle of a loop, but you can exit mid-loop.
- You cannot have an IF statement attached to a LOOP or END_LOOP. If you want a LOOP done conditionally, use a GOTO before it to control the skip.
- Use a >REPEAT if you want data for a certain response to be in a particular place.

- Loops do not work with CAT or FLD HIDE questions because loops get the response code from the answer array and hidden questions do not put answer in the answer array.
- You can also use the LOOP block to spread out the response codes to a previous multiple-response question without having any questions within the LOOP block. You may want to do this for data processing purposes.

```
EX:
{STATES:
Enter the western states you have visited
!CAT,,20
AZ Arizona
CA California
.
.
WY Wyoming
}
{
!LOOP,States,20}
{
!END_LOOP}
```

This would put the response codes AZ, CA, etc. in the data (in addition to the punches collected on the STATES question). You could then use these codes later for data processing. You can also capture the order the responses were entered this way, if you wanted to save that information. The multiple response FLD question also lets you do this, but has stricter limits than LOOP does.

When a NUM question or EXP statement is used as the control question, the response determines how many times the LOOP will be executed up to the maximum number of times through the loop. The program will store a 1 for the first loop controller, 2 for the second, etc.

Here is a sample LOOP/END_LOOP block with a NUM question as the control:

```
EX:
{CHILDREN:
```

```
How many children do you have?
!NUM,,,1,10,,NA }
{
!IF [CHILDREN$]="NA"
!GOTO,HOME}
{
!LOOP,CHILDREN,10 }
{NAME:
What is his/her name?
!VAR,,15,1 }
{AGE:
How old is \:NAME:?
!NUM,,,1,18,,NA,DK }
{SCHOOL:
What school does \:NAME: attend?
!VAR,,20,1 }
{ENDKIDS:
!END_LOOP }
```

How many times the questions in the LOOP block (NAME, AGE, and SCHOOL) are asked depends on the numeric response to the control question (CHILDREN) up to the maximum number of times specified (10). If a respondent says he has five children, the LOOP will be executed five times. Note the back-referencing of the child's name within the loop. Each time through the loop, the program will substitute the name of the latest child.

The LOOP in the above example uses 39 columns for each child, 2 to store the number of the child, 15 for the name, 2 for age, and 20 for the school; the total number of columns allocated is 390 (39 X 10). The start of each loop iteration is printed in the CHK file.

TIP:

For resumed interviews, put a {!RESUME_HERE} statement before each !LOOP question and {!RESUME_WHERE_AT} after each {!END_LOOP} statement.

This guarantees that all iterations of a given LOOP will be re-executed when the interview is resumed. In addition, !LOOP statements can be placed within !RESUME, !SUSPEND, !SPECIAL and !AFTER_QUIT blocks.

To see the current value of the loop controller during the loop iteration, if the controller is a CAT or FLD question then refer to it, but if the controller is a NUM or EXP question, refer to the LOOP question itself.

3.1.5 System Information Statements

Quota Incrementing Statements

Quotas are counters that are kept across interviews (on a stand-alone PC), and across interviewers (in a shared files environment). These are usually used to determine whether to continue an interview or not based on a set of questions or information collected at the beginning of the interview. The quotas can also be used just to keep track of counts during the interviewing period. The quota values can be seen or modified using the QUOTAMOD (see *5.5 QUOTAMOD)* and SURVSUPR programs (see *4.4.2 SURVSUPR MAIN MENU)*, or they can be written to the data or phone file to be used later in reports. (Also see the QUOTARPT batch file in \CFMC\GO).

Quota, or QUO, statements cause a counter in the QUO file to be updated. The QUOTA function will tell you the value of the named quota at the start of the interview. The current quota value can be obtained using the QUOTA NOW or QUOTN functions (see *2.6.4 USING FUNCTIONS IN CONDITION STATEMENTS, Response Counting Functions*). The READ_NAMED_QUOTAS compiler command will get you the current value of all named quotas. The MODQUOTA and MODQUOTN functions can be used to determine how often a particular quota has been updated in an interview.

The syntax for this quota statement is:

```
!QUOTA,subtype,quota name or location of name to increment,amount to increment,NOW
```

Subtype 1 is the default; this updates standard named quotas. Subtype 2 is used to update numbered quotas.

For **subtype 1**, the quota to increment is a quota name or a question label or data location that stores a quota name to be updated. If it is a question label or data location, it must have ()s around it.

Quota names may be up to 16 characters long, using alphanumeric characters, periods (.), or underscores (_). Quota names must begin with an alpha character, except they cannot begin with QQ to avoid confusion with question number references. They also cannot end with a period followed by numbers only; this is to avoid confusion with data location references. If using triplequota mode (described later), quota names are limited to 14 characters since .R and .T will be added to the name to designate the resettable and target quotas.

For **subtype 2**, numbered quotas, the quota to increment refers to the quota *number* to increment. It can be a specific number, or a question label or data location which contains the number of the quota to increment. More information on numbered quotas is discussed later in this section. You can have both subtype 1 and subtype 2 quota statements in the same interview.

The quota increment is usually 1, if counting interviews, but it can be any number (including a negative number to decrement), or a question label or data location containing the number to increment the quota with. Zero increment of quotas is allowed. The increment can also be specified as a particular number to change to by preceding it with an equal sign (=).

```
EX:
!QUOTA,,OREGON,=10,NOW
```

When specified in this manner, the comma between the quota name and the value is optional (i.e., OREGON=10).

The NOW option would cause the quota to be incremented immediately. This may be necessary for long interviews (so that new interviews will not start in the interim), or if you wish to update a quota prior to an interview being suspended. The default is that the quota is incremented when the interview is completed and a record is written, regardless of when the quota statement was executed. This is to guard against quotas being updated for interviews that may end up being terminated. Therefore, use the NOW option with caution.

If the interview is aborted without writing a case, standard quota updates are dropped, with only QUOTA NOW updates being done.

```
EX:
!QUOTA,,NEW_YORK,1
```

This example is using subtype 1 (the default, since none was specified). The quota named NEW_YORK will be incremented by 1 at the end of the interview if the data is saved.

You can also reference a data location where the quota label is stored in the quota statement. In this way, you can write one statement that says requests an update of the value of the quota found in a certain location instead of one statement per each possible quota updated. This is similar to the way numbered quotas work. The syntax to do this is:

```
{!QUOTA,1,[<label or location>],+1<,now>}
```

“Label or location” is the place where the quota name is stored. This works particularly well when your quota name is based off the market name or it comes from the sample file. For example:

```
EX:
{mkt: !phone,g,51,4}
{qname: .5
Q\|mkt
!spc,9}
body of questionnaire
body of questionnaire
body of questionnaire
{!quota,1,[qname],+1}
```

In this example, the quota name is derived from the market name, and it is updated by pointing to a !spc,9 question which loads the quota name based on the market field. If you put a reference in brackets ([]s), it knows to check the data location and use that label. If you have a label only, it assumes that is the name of the quota.

Note: Use Markets and the PHONE,M statement to keep sample that is over quota from being picked up by the interviewer. See *6.6.4 MARKET AREAS* for more information.

Techniques for Using the Quota System

There are three ways of specifying quotas.

1 If you only have a few quotas, you would most likely use specific references to standard quota names. These are easy to read and easy to keep track of. They have the disadvantage: If there are many of them, they need a separate statement for each quota check. Also, you must recompile the questionnaire when you have target quota value changes during interviewing. You can have up to 3,000 standard quota names.

2 The second way to use quotas is by using standard-named quotas in Triplequotas mode. This uses three separate counters for each quota: the quota value itself, a target quota (the number you are trying to reach) and a resettable quota. This use of quotas only allows 1,000 quotas, but has the advantage of ease of readability and easy execution of changes in the target values, without recompiling the questionnaire. (A variation of this would require using the SET_QUOTA compiler command to have only some of the quotas in this multiple mode. This allows you access to all 3,000 quotas.)

3 For more than 3,000 quotas, you must use numbered quotas. These are referenced by number, not by quota name. You can have up to 2,000,000 numbered quotas. If properly written, you can update all your numbered quotas with as few as one to five statements. The disadvantage: numbered quotas are more difficult to set up and track. You will probably have to develop your own quota list to be used for reference by the supervisor(s) and/or interviewers.

Creating the Quota (QUO) File

When the specifications for a study are compiled by PREPARE, the file <studyname> with a QUO extension will be created in the local directory or group unless you have -QUOTA_FILE in your header statement. If there is an existing QUO file of the same name, PREPARE adds any new quota names to that file; any existing quota values are left alone. A quota is created for each quota name and initially set to 0 (or the target value, if using Triplequotas mode and the TARGET compiler command) or to the value specified on the SET_QUOTA compiler command.

If you add new quotas or rename old ones, the old quotas retain their values. To get rid of unused quotas, delete the old quota file before rerunning PREPARE; on the other hand, you should be careful of removing it if you want to retain the existing values. When recompiling with added quotas, copy the QUO file locally so it will get the changes made to it. Then you can move it back to where it came from. If you forget to move it locally, a brand new QUO file will be created. If you recompile and make a new QUO file with new or differently named or ordered quotas, you cannot use the old quota file for interviewing. Use the QUOTAMOD utility's OUT and IN options to read the old value(s) in.

Using Named Quota Statements in the Questionnaire

A questionnaire that uses quotas is generally set up in the following sequence:

4 Data entry, generation, or PHONE questions get initial information for quotas

5 Condition(s) to check for full quotas

6 Statement(s) to display information about full quotas, and end the interview

7 The body of the questionnaire

8 Statement(s) to increment the quotas

With this sequence, the quota is checked before the body of the interview is conducted. If the quota is full, you could display which quota is full and end the interview. If the quota is not full, the interview is conducted and the quota is incremented.

An example using standard quotas follows.

The number of respondents desired is 1000, 250 from each of four cities. CITY is the control question. The next question checks for any full quotas and terminates the interview if there is one. Following the interview are the quota-incrementing questions.

```
EX:
''Check for quota
{CITY:
Which city are you calling?
!CAT

1 New York
2 Chicago
3 Atlanta
4 San Francisco }
''Continue the interview if the quota is not full
{
!IF (City(1) AND QUOTA(New_York)<250) OR &
(City(2) AND QUOTA(Chicago)<250) OR &
(City(3) AND QUOTA(Atlanta)<250) OR &
(City(4) AND QUOTA(San_Francisco)<250)
!GOTO,Intrview}
''Display quota that is full
{
Thank the respondent and report that the
\I\:City:\E QUOTA is FULL.
!DISPLAY}
''Terminate, don't do the interview
{
!QUOTA,1,OVER_QUOTA,1,NOW}
{
!SPC,B}
{Intrview:
.
. BODY OF INTERVIEW
.
''Increment quota at end of completed interview
{
!IF City(1)
!QUOTA,,New_York,1}
{
!IF City(2)
!QUOTA,,Chicago,1}
{
```

```
!IF City(3)
!QUOTA,,Atlanta,1}
{
!IF City(4)
!QUOTA,,San_Francisco,1}
{Endit:
!GOTO}
~END
```

At the end of the interview, the appropriate quota(s) is (are) incremented. If the interviewer backs up over the quota incrementing statements, the quotas will be de-incremented. You want to keep the increment statements at the end of the questionnaire typically to ensure that the respondent completes the interview.

You can increment as many quotas as you like in an interview. A separate quota statement is entered for each quota, with a relevant condition.

The quota will be updated when the interview is completed and the data case is written to disk, unless you use the NOW option on the quota statement, in which case it will be updated immediately. This was used on the quota OVER_QUOTA in the example above, since we were then aborting them; if we didn't use NOW on them, their quota value would never change since they write no data to disk. All quotas not incremented NOW store the increment in memory until the data case is written to disk. The MODQUOTA and MODQUOTN functions will give you information on quotas not incremented NOW.

Notice the QUOTA functions on the !IF statements above. The values picked up with the QUOTA function are the values of the quotas at the beginning of the interview. The QUOTA function is not affected by any QUOTA statements within the specifications. Use the READ_NAMED_QUOTAS compiler command to see the current value in the shared systems.

Using Triplequotas Mode

Triplequotas mode creates a resettable quota and a target quota for each quota name you reference in the questionnaire. To use this mode, you must specify the TRIPLE_QUOTAS option on the header statement and use the TARGET compiler command at the beginning of the interview.

In this case, three quotas are created for each quota reference. For the quota NEW_YORK, NEW_YORK, NEW_YORK.R (meaning resettable), and NEW_YORK.T (meaning target) are created. Each time NEW_YORK is incremented, NEW_YORK.R also will be incremented. You can then reset NEW_YORK.R if you want to compare the current quota value with its value at an earlier time in the study. You can use the target quota to change the value of the quota you are using as a target. In this case, you would want to write your quota checking statement to compare the quota and its target quota. Here is an example using Triplequotas mode:

```
      EX:
[Ctest,,"Test of product markets",TRIPLE_QUOTAS]
{!TARGET New_York = 250}
{!TARGET Chicago = 250}
{!TARGET Atlanta = 250}
{!TARGET San_Francisco = 250}
''Check for quota
.
{
!IF (City(1) AND (QUOTA(New_York) < QUOTA(New_York.T)) OR &
(City(2) AND (QUOTA(Chicago) < QUOTA(Chicago.T)) OR &
(City(3) AND (QUOTA(Atlanta) < QUOTA(Atlanta.T)) OR &
(City(4) AND (QUOTA(San_Francisco) < QUOTA(San_Francisco.T))
!GOTO,Intrview}
''Report full quota, go to end, don't do the interview
{Intrview:
.
. BODY OF INTERVIEW
.
''Increment quota
''End interview
```

The TRIPLE_QUOTAS header option causes the quota names .R and .T quotas to be made. The !TARGET commands set the initial target value for the .T quota. It is not necessary if you want to set these yourself using QUOTAMOD or SURVSUPR. The target quota can be changed at any point after this, also. With this scheme, you would not want to ever change the quota value, just the .R and .T quota values. You also do not have to recompile your questionnaire to change target quota values this way.

You cannot use the same QUO file if you have compiled with Triplequotas on and try to recompile with Triplequotas off (or vice versa).

TRIPLE_QUOTAS keyword now has a new parameter "Modify_targets_only".

Here is the syntax:

```
Triple_quotas=modify_targets_only
```

If you put this in the questionnaire header, the program will only allow interactive modification of the TARGET values, not the actual quota values or resettable quota values. This will keep supervisors from mistakenly changing the actual counts on a job.

NOTE: Use the “RESET” command in quotamod to reset the resettable quotas when this command is in effect.

Master Quota File Helps Control Multiple Studies

You can count quotas across multiple studies using the Master quota" feature set that is described below:

SYNTAX:	**FUNCTION:**
!quota,R,<name>,# !quota,REGULAR,<name>,#	Update regular quota
!quota,N,<name>,# !quota,NUMBERED,#	Update numbered quota
!quota,MR,<name>,# !quota,MASTER,<name>,# !quota,MASTER_REGULAR,<name>,#	Update MASTER quota
!quota,MN,<name>,# !quota,MASTER_NUMBERED,<name>,#	Update Master Numbered quota
!spc,a,N	Get “unique number” from regular quota file
!spc,a,M	Get 'unique number' from master quota file

!expr,,xf(MASTER_QUOTA,<name>)	Return master quota value for <name>
!expr,,xf(MASTER_QUOTA,<name>,MODIFY)	Return the amount the quota was modified since beginning of the interview for <name>
!expr,,xf(MASTER_QUOTA,[label])	Return the value of the master quota named in the location of [label] in the data file
!expr,,xf(MASTER_QUOTA,[label],MODIFY)	Return the amount the quota was modified since beginning of the interview for master quota named in [label] location.

To tell the study the name of the "master" quota file, in the questionnaire header type:

```
[study,masterquota=study2]
```

And compile "study2" to build the quotas into it:

```
 ~prep compile
[study2]
{!setquota name1 = 0 }
{!setquota name2 = 0 }
{!setquota name3 = 23 }
~end
```

Using the SET_QUOTA Compiler Command

Sometimes you want the capability of Triplequotas mode, but don’t want every quota to be in this mode, since every quota now becomes three quotas. You can get around this by using the SET_QUOTA compiler command. This command lets you set the value for quotas individually, so some quotas can be in single quota mode, others can be in double mode, and still others can be in triple mode.

You do not need (and do not want) TRIPLE_QUOTAS on your header statement. You can call the pieces of the quotas anything you want; you are not restricted to name.R or name.T.

```
EX:
{!SET_QUOTA males.tar=50}
{!SET_QUOTA females.tar=75}
{GENDER:
Indicate gender of respondent
!CAT
1 Male
2 Female }
{
!IF (GENDER(1) and QUOTA(males)<QUOTA(males.tar)) or &
(GENDER(2) and QUOTA(females)<QUOTA(females.tar))
!GOTO,Q3 }
```

Using Numbered Quotas

Numbered quotas are updated with the quota subtype 2 statement. Since the number of the quota to update can be constructed and placed in the data, you only need to have one (or a few) quota update statements to access all of your quotas. Also, while you can have up to 3,000 named quotas, you may have many times more numbered quotas. You must specify the highest number you will be using for your numbered quotas with the MAXIMUM_QUOTA_NUMBER option in your Header statement. The theoretical limit for numbered quotas is 2,000,000 but in NO WAY is it recommended that you create a quota file of this size because the resulting quota file would take up 16 gigabytes of disk space.

The easiest way to use numbered quotas is to set up a numbering scheme so you can easily determine which quota cell you are working on. Disk-based FLD questions are often used as the structure to capture the numbered quota number to be updated.

Suppose you have a quota scheme that is sex by three age groups by each of the 50 states. This will produce 300 quota cells (2 x 3 x 50). You can use the two digits of the number for the state reference (codes 01 thru 50 for the 50 states), use a digit to tell you which sex (1 or 2), and use a digit to tell you which age group (1-3). You can also set up target quotas for each quota by adding a digit on the front (e.g., 1 for quota, 2 for target).

By storing these codes in consecutive data positions, you can use a disk-based FLD question, EXP statement, or data location to reference this constructed quota cell number, e.g., !QUOTA,2,[9.5],1, if the quota type (quota or target) was in column 9, state codes were in columns 10-11, sex in column 12, and age group in column 13. Using this scheme, the quota for Alabama, male, young age would be 10111, and its target would be 20111. The example below shows how to collect, store, check, and increment the numbered quota cells.

```
EX:
''Get quota related info (as numbers in
consecutive columns)
{State:
ENTER A 2 DIGIT NUMBER ASSOCIATED WITH STATE
!FLD
01 ALABAMA
02 ALASKA
...
50 WYOMING}
{Sex:
ENTER SEX OF RESPONDENT
!FLD
1 MALE
2 FEMALE}
{Age:
ENTER AGE OF RESPONDENT
!FLD
1 18-35
2 36-54
3 55-70}
''Set up a question to hold the numbers as the
'quota number'.
{QuoInfo: [State.4]
!VAR,A,4,4}
{Numquo: .5
!EXP,,[Quoinfo]+ 10000}
{Targquo: .5
!EXP,,[Quoinfo]+ 20000}
''Display the quota if it's full and go to end
{
```

```
!IF QUOTN(Numquo) >= QUOTN(Targquo)
QUOTA number - \:Numquo:
FOR SEX=\:Sex:
AGE=\:Age:
STATE=\:State: IS FULL!!!
Thank the respondent and Press Enter to continue
!DISPLAY}
{
!IF QUOTN(Numquo) >= QUOTN(Targquo)
!GOTO,Endit}
.
. Body of questionnaire
.
''Don't need an !IF statement because you will
always update
''SOME quota here
{
!QUOTA,2,Numquo,1}
{Endit:
!GOTO}
~END
```

This example gets the relevant information, then prepends a 1 or 2 to the four-digit number to reference the quota and target (e.g., 12123 and 22123) using EXP statements. Then the QUOTN function is used to compare the values of the quotas specified. The same kind of logic is used for the rest of the questionnaire as for standard quotas. Note that only one QUOTA statement is needed since we can generate numbers to reference the correct quota. Standard quotas require a conditional quota update statement for each named quota.

Numbered quotas do not affect the normal operation of standard quotas, so you can have both in the same questionnaire.

Numbered quotas also understand a specific quota number specified as the quota to increment.

```
EX:
!QUOTA,2,1234,5
```

Updating the Quota File

Survent will not start up unless the QUO file matches the current QFF file. That is, if you have added quota references, deleted quota references, changed the name or changed the order of your quota references in the spec file, and you continue to try and use your old quota file, you will get a blow error. You must recompile, updating the current QUO file, or else use the QUO file you made when you last compiled.

Survent saves quotas as positional items. If you made any of the above changes you would most likely cause the program to update the ***wrong*** quotas, which would also be very difficult to figure out. This feature prevents that problem from occurring.

When you PREPARE a spec file, a CRC check (cyclic redundancy check) is done on the names block of the quota file built, and is stored in the QFF file. When you go to run Survent, the CRC value in the QFF file must match the value in the QUO file you use, or the program will blow.

The TIMESTAMP option on the header statement, if used, means you ***must*** use the quota file you build when you compiled when you start Survent. A time stamp is put in the QUO file and the QFF file when you compile, and Survent will compare them and only let you continue if they match. This is to prevent people from compiling a questionnaire and loading it to be interviewed without updating the quota file at the same time.

NOTE: If you do not use the TIMESTAMP option, the program still does a CRC check to assurethat the quotas you are using are correct, it just does not verify that the quota file was the exact one you compiled the questionnaire with.

When adding new quota names to an active study, there are two procedures that will make valid quota files, and two that will make an invalid quota file.

To make a valid quota file, do one of the following:

9 During a break in interviewing, copy the current updated quota file to the directory you compile the new questionnaire in, and re-compile the questionnaire. Then, copy the updated quota file back to the interviewing directory. This will cause the new quotas

to be added and the questionnaire file to properly address the quotas.

10 Compile the new questionnaire. Run QUOTAMOD and use the OUT option to extract the current values from the in use quota file. Move back to the directory you compiled the questionnaire in, run QUOTAMOD and use the IN option to add the quota values to the quota file with the new names in it. Copy this quota file to the interviewing directory and re-start interviewing.

Special Information Statements

Special Information, or SPC, statements are used to share information with systems outside of the interview, or to perform special functions during the interview.

The syntax for a Special Information statement is:

```
!SPC,subtype,options
```

Most of the subtypes have no options. The exceptions are noted.

NOTE:

- Please note that several SPCs give you more than one type of information. If you want the third piece of information out of five possible pieces, you have three choices. You can ask for three pieces and ignore the first two, you can look to see if another SPC gives you the same information in a more reachable position or you can specifically request certain fields (SPC,5 and 7).
- SPC subtypes with a TO location and FROM location will always have the data location and length specified on the !SPC line, not on the question label line.
- The location in the data record may be referenced as absolute column, card/column, QQ#, or label. (See header statement option SPL_DATA_LOCATIONS_OK for an exception.) Brackets around column or card/column references are optional. They are included below to make it easier to note which option is the location.

- 3000 is the maximum length you can get or put at one time; to work with more, use an additional SPC. The SPC statement subtypes are explained below. **SPC,3** puts the date and time from the system into the data in one of two formats. In versions 7.6+, or if you specify **SPC,3,L**, it uses the format: YYYYMMDDHHMMSSXJJJ (18 characters), which is year, month, day, hour, minute, second, day of week, and Julian date. For instance, '200003071320542066' would be returned for March 7, 2000 at 1:20:54 P.M., which is a Tuesday and the 66th day of the year. For days of the week Sunday is 7, Monday is 1, …, Saturday is 6.

The default width is 18 columns; you may specify less. To control the length of the field, if overriding the default, use the question label line.

```
EX:
{.12
SPC,3}
```

This would put the current year, month, day, hour and minute in the data. The data from this statement is in the format necessary for the TIMEDIFF function to figure out the difference between two date/times.

TIP: Use a column offset to reference specific columns in this data field. Refer to the example in section *2.6.1 REFERENCING QUESTION TYPES, CAT and FLD Question* Types.

```
EX:
!IF [DATE+4.2] = 10
```

This means count over four columns from the beginning of the data field identified as DATE (an SPC,3) and if the next two columns holding the date equal 10 then this condition is true.

SPC,4 enters the date and time of the interview in the text format "DDD MMM DD YYYY HH:MM xx"; for example, "THU NOV 10 1994 3:38 PM". The default width is 24 columns (19 characters and 5 blank spaces); you may specify less. The amount of information entered into the data is specified on the question label line as the length.

NOTE: Time is machine time when the SPC,4 is executed.

```
EX:
{.15
!SPC,4}
```

SPC,5 stores information about the interviewer from the employee information file. The syntax for the SPC,5 question type is:

```
!SPC,5,start col,length
```

What you get depends on the starting location and length you specify. The available information is:

1-4	Interviewer ID
5	blank
6-30	Interviewer name
31	blank
32-40	Special types (1-9)
41	blank
42-45	B=can Abort, break or Change interviews C=can Change interviews D=Debug mode (overrides interviewer controls, allows GOTO forwards and backwards, can Abort, Break or Change an interview N= Interviewer has no capabilities and cannot view prior interviews E=interviewer started in Echocats mode by default L=do interviewer logging and save at end of interview Q=do “debug mode” interviewer logging and save after every question
46-240	User-supplied employee information

```
EX:
{
!SPC,5,6,35}
```

This example will get you 35 columns of information starting in column 6 (Interviewer name and special types).

!SPC,5,42,4 will return the interviewing modes in use RIGHT NOW; so, for instance, if you sign on to an interview in TRAINING mode (either by the supervisor or saying "intv,t" when logging in, you will see a "T" in column 42-45 even though there is no "T" in the employee.xxx file, columns 42-45.

NOTE: This is the same layout as in the Employee information file (*4.4 INTERVIEWING WITH Survent, Employee Information File)*. If the interviewer is a special type, regardless of how that was set, a 1 will be in position 32 (as noted above) is they're special type 1, a 2 if they are special type 2, etc. If you are using the "Long_Employee_ID_Location: ##.##" parmfile parameter, the long name must be specified in columns 46-240, and will only be recorded in the data if you specify and !SPC,5 statement to store it.

SPC, 6 enters time (in seconds) since the beginning of the interview. The default width is 9.

Control the width, if overriding the default, on the question label line.

```
EX:
{.6
!SPC,6}
```

This example would enter the time since the beginning of the interview in a six-column field. SPC,6 is often used to time all or part of an interview. If using for this purpose, remember to account for any time spent suspended. Also see SPC,Z, which lets you account for suspend time and get partial times without calculations.

SPC,7 puts the following information in the data:

```
0        1         2         3         4         5         6         7         8         9         0         1

123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
--------------------------------------------------------------------------------------------------------------
DBUGA0305291444 ??!00001 SPC7L    0         06 04 0000 0305291443
2003052914440000 000
DBUGA0305291444 ??!00001 SPC7L    0         06 04 0000 0305291443
```

Information:	Number of Characters:	Relative Start Position:	Information taken from:
Interviewer ID	04	01	Employee info file/Login
Interviewing Mode	01	05	CfMC system (A=standalone B=under boss or S=Server)
Date/time	10 (MMDDYYHHMM)	06	Local operating system
Blank	01	16	
Site code or Dialer #	02	17	Site code if used (DOS only), otherwise Dialer #
Short language	01	19	Current language in Effect (1 char. only).
Station number or Ldev	05	20	from TTYINFO file or DOS CFG file
Blank	01	25	
Study name	08	26	from QFF file or DOS CFG File
Blank	01	34	
Special interviewer type	09	35	from Employee info File or Login
Blank	01	44	
Process time zone	02	45	from TTYINFO file
Blank	01	47	
Terminal type	02	48	from TTYINFO file (HP, UNIX only)
Blank	01	50	
Phone extension	04	51	from TTYINFO file
Blank	01	55	
Qff file compile time stamp	10 (YYMMDDHHMM)	56	from QFF file
Blank	01	66	
Number of backups	03	68	Times backed up in Survent
Maximum consecutive backups	03	72	Max. # of consecutive questionnaire backups
Blank	01	75	
TTY file device info	20	76	TTYINFO file columns 53-72
Blank	01	96	
4-digit year datetime	12 (YYYYMMDDHHMM)	97	Local operating system
2-character language code	02	110	Current language in effect

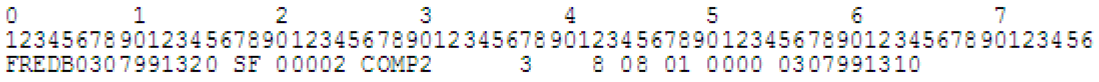
The maximum length of !spc,7 is 110.

The Interviewing mode can be "A" for standalone interviewing, "B" if running under a supervisor and "S" if running under the CfMC server without a supervisor. The language can be whatever is specified in the LANGUAGE= parameter in the study header.

Special interviewer types may be numbers from 0 to 9, and are listed in position, starting in column 35. Time zones may be 01-24 (U.S. mainland time zones are 05-08). The terminal types (columns 48-49) are:

01	ANSI
02	Dialup
03	Unknown
04	Vt 220
05	Wyse 50
06	Wyse 150
07	Wyse 85
08	IBM 3151
09	HP
10	Adm 3
11	Adds
12	HP 700/41
13	Att 705
14	HP 700/96
15	HP 700/98
16	Wyse 60
17	Vt100
18	Ansi25
19	Scoansi
20	At385
21	HPtelnet
22	Linux
23	Xterm

The phone extension will be "0000" unless you are using a dialer or SoundSurvent, which use the extensions. Here is an example of the data returned:



```
0        1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890123456
FREDB0307991320 SF 00002 COMP2      3    8 08 01 0000 0307991310
```

This shows that the interviewer is FRED, the interviewing mode is under a supervisor ("B"), the date/time is 03079941320 (March 7, 1999, at 1:20 PM), the site code is SF, the station number is 00002, the study name is COMP2 and the special interviewer type is 3 and 8. The time zone is 08 (pacific time), the terminal type is 01 (ansi), the phone extension "0000", and the compile date/time is 0307991310. The compile date/time is especially useful to track questionnaire versions. Use SPC,7,L to get a 4-digit year datetime string in position 87 to match version 7.6l dates.

Here is the syntax for the SPC,7 question type:

```
!SPC,7,start col,length
```

The default width is 86 (all of the above information), but a width is required either by using the length parameter above or specified on the question label line.

```
EX:
{
!SPC,7}
```

This example will get you all columns of the information. See *4.1.1 MODIFYING THE CONFIGURATION FILE* for more details on the items reported by the SPC,7.

SPC,9 causes text on the text line of this question to be saved in the data and as a response in the answer array. Text controllers such as back-references may by used to fill in values (i.e. text of a response from a prior question using \:label:). If you have multiple text lines, the nth line (n being determined by a controller) of text on this question will be saved. The default width is the longest line of text; the maximum width is 2100. You can change this width on the question label line. Making this width smaller than the longest text line will only affect the information stored in the data, not the text available for back-referencing.

Here is the syntax and an example for SPC,9 question type:

```
text line 1
...
text line n
!SPC,9,controlling label or location }
```

SPC,9 statements require one or more lines of text and can use a controller question label or location. If no controller is specified, then you can have only one line of text (first line) and it will be what is saved. This is the ONLY way to get text stored in the data, there are no GEN statements or other statements to do so.

```
EX:
{NEWTEXT:
Put this in data
!SPC,9}
```

Here is an example of storing the text of a response to a single response CAT or FLD question in the data. Notice the use of the length parameter as the program will assign a length based on the text on the back-reference otherwise (in this example, it would assign “11”).

```
EX:
{TEXTRESP: .30
\:typelike:
!SPC,9}
```

The SPC,9 statement will save the nth line of its text, corresponding to the number stored in the control question.

```
EX:
{SAVETEXT:
first
second
third
fourth
fifth
!SPC,9,TIMES }
```

```
{
What did you do the \:SAVETEXT: time you ate
there?
...
```

This SPC,9 statement will check the numeric response to the question TIMES and save the line of text corresponding to that number. If the response to TIMES is 3, line three (third) will be saved. At a future question you could then back-reference the text line stored.

NOTE: The SPC,9 statement requires a specific width if it sees back-references or a backslash in the text. This is because if you don't, the statement is assigned a width based on the text it sees, not on the "filled in" text from the backslash (eg. \^5c) or back-reference (eg. \: or \|).

SPC,A assigns the case ID now, if it is not already assigned, rather than at the end of the interview, and places it in the data in the specified case ID field (see header statement). This is particularly useful should you want to give suspended interviews the proper start sequence number; otherwise, they would be assigned an ID after the interview is completed, perhaps many days later.

Using this may cause gaps in the ID sequence, if, for instance, an interview is terminated (see SPC,B below) after the ID has been assigned. You should probably not have a condition on the SPC,A statement; you would want to do this always or never. No data location or width can be specified on the question label line, the data is saved in the case ID field, which has a width already determined or specified.

```
EX:
{
!SPC,A }
```

Another use of the !SPC,A is to generate a sequential number from the quota file independent of the case id. You must specify a location where the data will be placed. This will allow you to get a unique number for each case. It could for instance, be used when you are only assigning case ids to completed interviews but saving other data, or when you are editing data or using! CALL questionnaires where there are multiple data records generated per case id.

```
EX:
{ .5
!SPC,A,N }
```

If you'd like, you can specify the length on the !SPC line using the syntax !SPC,A,N,#, where # is the length.

The value is saved in the quota file and may be accessed using QUOTAMOD's "OUT" option; the exported ASCII file will have a line that reads: "*unique_number=23" something similar. You can modify the value by changing it and taking the "*" off the front of the line similar to the way you treat times reported, and reading the file back in using the QUOTAMOD "IN" option.

SPC,B causes the interview to terminate, without saving the data. Use caution with this, as you may want to save the data from terminated interviews for counts, etc. SPC,B statements should be controlled by a GOTO statement (skipping to or around them) or IF condition. No data location or width can be specified on the question label line.

```
EX:
{
!IF OVERQUOT(YES)
!SPC,B }
```

!SPC,B,O allows you to to abort from a !CHILD questionnaire and return to the parent or "!SPC,B,ONLY_CHILD". Previously, if you aborted out of a child questionnaire you could not return to the parent. You could only abort completely out of the questionnaire.

SPC,D sets the case ID to the data found in the columns specified for the case ID in the header statement. No data location or width can be specified on the question label line.

```
EX:
{
!SPC,D }
```

SPC,E sends messages to the SURVSUPR or any monitors. This is useful for informing the supervisor or monitor when an interviewer has reached a certain point in the interview or alerting the supervisor when a quota is completed. The question text will *not* appear on the interviewer's screen.

Here is the syntax and an example for the SPC,E question type:

```
text to be sent to survsupr/supervisor
DAI/monitorlist/log record

!SPC,E,option

EX:
{
Supervisor—San Francisco quotas are full!
!SPC,E
}
```

The option is not required; the default is option 2.

Option 1 causes up to 14 characters of text to appear on the supervisor's DAI screen or the 72 characters to any monitor's interviewer list for the interviewer sending the message. The message from option 1 will also be sent to the server's log file. Each subsequent !SPC,E,1 overwrites the previous one displayed on the supervisor's DAI list or the monitor's interviewer list. This is most often used to keep track of interviewer position in the questionnaire.

```
EX:
{
Screener completed
!SPC,E,1 }
```

Option 2 sends the text as a specific message to the supervisor when they are in Run mode. The screen is paused as the message displays. There may be up to nine 72 character messages per SPC,E,2. This is most often used in special notification situations, such as when quotas are full (see SPC,E example above).

Option 3 sends up to 40 characters of information to the LL14 record in the server LL log file. You can use to send the date/time of the start of the main interview for interviewer log reports, or to pass warnings to the log file.

```
EX:
{
This is to be passed to LL14 record
!SPC,E,3}
```

SPC,F puts the station number or LDEV of the interviewer into the data. This can come from the LDEV variable (DOS or UNIX) the TTYINFO file (UNIX) or the CFG file (DOS standalone) (also see SPC,7). Control the width, if overriding the default of five, on the question label line.

```
EX:
{
!SPC,F}
```

SPC,H aborts an interview or writes it to disk with a special case ID. No data location or width can be specified on the question label line.

Here is the syntax for the SPC,H question type:

```
!SPC,H,option
```

You may specify an option after the subtype H:

0 Default. Aborts the interview (like SPC subtype B), but then starts the next interview without stopping at the standard “Enter **Enter** to Interview” prompt.

1 Writes this case to disk with a case ID of NOTKNOWN (or as many of these letters as will fit in your case ID field). Doesn’t update any quota cells, and doesn’t update completes in SURVSUPR’s status screen. Any quota changed by a QUOTA NOW will still change the cell value. See SPC,W to write with a case ID.

2 Do not allow SUSPEND from the time this question is executed until the end of this interview or until an SPC,H,3.

3 Undo an SPC,H,2; allow SUSPENDs again.

4 Disable ^ from here on (until SPC,H,5).

5 Re-enable ^ after SPC,H,4.

6 Disable RESET from here on (until SPC,H,7).

7 Re-enable RESET after SPC,H,6.

8 Disable RETAKE from here on (until SPC,H,9).

9 Re-enable RETAKE after SPC,H,8.

A Sets the Force_Into_Interview time to the number of seconds specified. This is the time after which an interview will automatically be restarted (and a message sent to the Supervisor) after the "Return to interview" prompt. This lets you control on a study-by-study or interviewer-by-interviewer basis the amount of "rest time" between interview starts, or force the time high if the interviewer is going on a break.

Note: SPC,H does not update quotas unless you specify "now" in the !QUOTA statement.

The syntax for SPC,H,A is:

```
!SPC,H,A,seconds
```

A setting of zero will turn off the automatic starting of interviews when the FORCE_INTO_INTERVIEW=## option is in use by the supervisor. A negative setting turns off the 'slacking off' message to the supervisor, but otherwise uses that number.

Set this in the supervisor using the FORCE_INTO_INTERVIEW command in SURVSUPR.

NOTE: The functionality of options 2-9 is better attained by using the compiler commands ALLOW and -ALLOW (see *3.2.1 INTERVIEW CONTROL COMMANDS*). The advantage of using SPC,H options is that you can do them conditionally. The disadvantage is that they are not undone if the interviewer backs up over them; the ALLOW compiler commands are.

```
EX:
{
!SPC,H,1}
```

SPC,J causes a pause of a specified number of seconds; the maximum is 60 seconds and the number can include decimal places (i.e., 1.5 for one and a half seconds). No data location or width can be specified on the question label line.

```
EX:
{
!SPC,J,10 }
```

This says to pause for 10 seconds.

NOTE: You can use the DISPLAY question subtype 3 to display a screen and pause for some number of seconds.

SPC,K puts information in the local scratch area. This is a place that holds information between interviews for a particular interviewer. You must specify the *to* location in the scratch area and the *from* question or location. This is often used to ask a question in the first interview of the session (e.g., are you interviewing about compact cars or luxury cars?), then controlling later interviews with that information without having to re-ask the question. The information is gone when the interviewer quits that interviewing session.

Here is the syntax and example for the SPC,K question type:

```
!SPC,K,to position (scratch area),from label
or location,length
```

The local scratch area position may be from 1 to 1000. The maximum length is 240 at one time. No data location or width can be specified on the question label line.

```
EX:
{
!SPC,K,1,[1/52],5}
```

This statement says to store the data from column 52 for a length of 5 in the local scratch area starting at position 1.

SPC, K statements are used in conjunction with SPC, L statements and/or the LOCALSCRATCH function. Once data has been stored with the SPC, K, the SPC, L could get the data and/or the LOCALSCRATCH function could be used to check its value. The local scratch area is also useful for CALL questionnaires to move data between the main and sub-questionnaires, and also for Coding mode, or multiple phone calls (interviews) to the same person.

SPC,L retrieves information from the local scratch area, which had been stored there with an SPC,K statement from this or a prior interview.

Here is the syntax and an example for the SPC,L question type:

> **!SPC,L**,<u>to label or location</u>,<u>from position (scratch area)</u>,<u>length</u>

The local scratch area position may be from 1 to 1000. The maximum length is 240 at one time. No data location or width can be specified on the question label line.

```
EX:
{
!SPC,L,SAVEDATA,1,5}
```

This says to get the information from location 1 of the local scratch area for a length of 5 and store it in the location of the question labeled SAVEDATA.

Note: For SPC subtype L, the SPC statement itself can be the receiving question in the data. Specify a label on the question so that you can back-reference it later. You may not refer to it by just the label; you must refer to it by location. Leave 'TO location' blank, but be sure to add the comma placeholder for the missing parameter.

```
EX:
{FONUM:
!SPC,L,,1,10
{The information in the local scratch is:
\|FONUM
!DISP,1}
```

This means that you no longer need to set up a VAR,A, CAT,A or FLD,A using the same data location as the SPC in order to back-reference information placed in the data using one of these statements.

SPC,M puts information in the global scratch area. This is a place in the quota file used to hold things other than quota type numbers, e.g., text to be displayed, a Yes/No switch, or other information. This holds across the whole system if you are using a network or mini-computer base. It acts like the SPC, K if using a standalone system, except it will stay in memory until specifically changed.

You must specify the *to* location in the scratch area and the *from* question or location. A supervisor might want to use this scratch area to store a message that displays at the end of the interview or to set a switch that determines the product asked about that day.

Here is the syntax and an example for the SPC,M question type:

!SPC,M,to position (scratch area),
from question or location,length

The global scratch area position may be from 1 to 40. The maximum length is 40. No data location or width can be specified on the question label line.

```
EX:
{
!SPC,M,1,[2/32],8}
```

This statement says store the data from record 2 column 32 for a length of 8 in the global scratch area starting at position 1.

SPC, M statements are used in conjunction with SPC, N statements. Once data has been stored with the SPC, M, the SPC, N could get the data and the value could be checked.

SPC,N retrieves information from the global scratch area, which had been stored there with an SPC, M statement in this or a prior interview.

Here is the syntax and an example for the SPC,N question type:

!SPC,N,to label or location,
from position (scratch area),length

The global scratch area position may be from 1 to 40. The maximum length is 40. No location or width can be specified on the question label line.

```
EX:
{
```

```
!SPC,N,MESSAGE,1,8}
```

This says to get the information from location 1 of the global scratch area for a length of 8 and store it in the location of the question labeled MESSAGE.

Note: For SPC subtype N, the SPC statement itself can be the receiving question in the data. Specify a label on the question so that you can back-reference it later. Leave ‘TO location’ blank, but be sure to add the comma placeholder for the missing parameter.

```
EX:
{FONUM:
!SPC,N,,1,10
```

This means that you no longer need to set up a VAR,A, CAT,A, or FLD,A using the same data location as the SPC in order to back-reference information placed in the data using one of these statements.

SPC,P gets a case for Survent to use, typically for cleaning or recoding in Survent. You can use CAT,A, FLD,A, VAR,A, and TEX,A questions to display the current data, then build new questions to key new data in. The program will continue to search for cases that meet the criteria of the IF condition on the SPC,P until there are no cases left with the condition. The program will search for cases one more time after it has reached the end of the data file.

Here is the syntax and example for the SPC,P question type:

```
!SPC,P
EX:
{WHICH:
Enter the case ID that you want.
!VAR,,4,4}
{
!SPC,P,,WHICH}
```

SPC,P may be used in two different ways; the first way is to retrieve cases by their particular case ID. You would use this method if you know the case you want to get and don’t want the program to present you with some random case based on data criteria.

The second way is to use a conditional statement to get cases based on criteria in the data, such as whether you have questions that need to be edited or coded.

See the INDEX file in %CFMC%Survent (DOS) or $CFMC/survent (UNIX) for examples that provide a more complete description of the features you can utilize in coding/updating questionnaires.

Here is the syntax for the case ID getting SPC,P question type:

```
{
!SPC,P,suboption,variable with case ID to get }
```

The default suboption is "1".

Suboption 2 will continue with the questionnaire rather than leaving you at the "Return to Interview" prompt if it does not find a case. Suboptions 1 and 2 will get a SURVBLOW error if you try to get another case when you already have a case you are working on in the interview.

NOTE: With SPC,P,2, by default, when a matching case is not found, Survent prints a message saying that no case was found. By using dump switch ">DUMP Y2" you can turn the message *off*.

Suboptions 3 and **4** are like options 1 and 2, respectively, regarding how they handle it when a case is not found. They are different in that if you have a case you are working on and go to retrieve another case in the same interview, they will write the case in hand back to the data file and then go and retrieve the new case you asked for. These are only used in very complex database update situations.

NOTE: With SPC,P,4, by default, when a matching case is not found, Survent prints a message saying that no case was found. By using dump switch ">DUMP Y2" you can turn the message *off*.

Suboption 5 ignores the fact that the specified case is checked out and gets it anyway. This lets you return cases that were checked out because interviewers/coders aborted with errors while the study was still live, and it lets you read records even if someone else is currently updating them.

NOTE: If you make modifications while another person currently has the same record and is also making changes, only the last modifications are saved, so use carefully.

Here is an example of the SPC,P to get a particular data case:

```
      EX:

[MYSTUDY,CASE_LENGTH=800,COMMENT="Codingqstnr",QFFNAME=codeit]
{GETIT: .4
Which case ID would you like to get?
Enter "NM" for No More.
!NUM,Z,,1-9999,,NM }
{
!IF [GETIT.2$]="NM"
!SPC,B }
{
!SPC,P,,GETIT1 }

... rest of coding questionnaire
```

The questionnaire header must have the same study code as the data file that needs to be coded.

It must also match in terms of case length and text start. Note the use of the QFFNAME option to write a questionnaire with a different name than the study being coded. You may run the "updating" questionnaire at the same time as the questionnaire that is being updated is running under the CfMC Server.

The SPC,P needs the case ID that it will get to be the exact characters and width of the case ID in the file, which is why in the example the field used is a NUM,Z statement which zero-fills the field. The program will look for the next case with a case ID that has the exact characters of the question specified.

The program searches for the case in the data file. If a matching case ID is found, it is retrieved for review and modification. If not, it returns an error message and puts you back at the "Return to Interview" prompt. Once a case is found, the data you recorded your request for a case ID in is replaced with the data from the case found. If you wish to save data created prior to getting the new case, use the LOCAL SCRATCH area. Then retrieve the data after the case has been found. If you type ABORT or use SPC,B to abort the interview, the modifications are not saved.

If you have a large data file to update, you can use the NUMCASES=# option in the questionnaire header to build the data file with a case ID directory in it. Then, when you ask for a particular case, the program can go directly to that case rather than searching sequentially through the file from the beginning (which it otherwise will do).

Using SPC,P to get Cases Based on Data Criteria

Here is the syntax for the SPC,P question to get cases based on data criteria:

```
{
!IF statement
!SPC,P,suboption }
```

Unlike other conditional statements, the conditional statement on an SPC,P applies to the prospective cases you want to get, rather than the conditions in the current interview. If a case is not found which matches the data criteria specified, an ERROR message is returned and you are returned to the “Return to Interview” prompt (if using the default SPC,P suboption).

This type of SPC,P is used to do automatic coding or editing of open ends. It searches the file sequentially for the next case that matches the data criteria. Survent will search around the END OF FILE back to where you last started your search before giving you the message that it has not found a case; therefore, you will usually want to use a marker in the file to tell it that you have already been there and edited the particular case.

Here is a simple example that allows editing of only one question:

```
     EX:
[MINE,CASE_LENGTH=800,COMMENT=”Coding qstnr”,QFFNAME=CODE]
{EDITDONE: 500 HIDE
!FLD
1 Edited the questionnaire }
{Q1: 200 HIDE
Original Question 1
!CAT
(9) 9 Other response}
{
!IF Q1(9) AND EDITDONE(<>1)
!SPC,P }
... rest of coding questionnaire
{Marker: 500
Do you want to mark this case as coded?
!FLD
1 Yes
2 No
}
```

The questionnaire uses the condition to find cases that have an "Other" response to question 1 and have yet to be coded or edited. All questions from the original questionnaire can be redisplayed and used for conditional statements once the case to edit has been found. You can have an option to pick which question to code, whether to code or edit open-ends, etc.

By specifying the case ID field you can get a particular case ID. Note that it may be collected as a number but it has to be evaluated as a string.

```
EX:
{GETIT: .4
Which case ID would you like?
Enter "NM" for "No More"
!NUM,Z,,1-9999,,NM }
{
!IF [GETIT.2$]="NM"
!SPC,B }
{GETIT1: [GETIT]
!VAR,A }
{
!SPC,P,,GETIT1 }
```

SPC,R sends a data record to the CfMC server to be written to a "savelog" file. This is typically used to log interview start/stop times and other information about the study. This file is specified in the parmfile by a line "sv_logging: <filename>, <record

length> where <filename> is any file to which you have write access. Here is the syntax for SPC,R:

```
    !SPC,R,label or location
```

The lable or location is where the data to send to the log file resides. If the file is not yet built, it will be created; if it does exist then records will be added to the end. The data being sent may be any length, but only up to <record length> bytes will be written to the file.

SPC,S copies from the data to the suspend comment. This is useful in a Suspend block so the suspend text can be obtained in an easier way. If used, the prompt for the suspend comment will not come up.

Here is the syntax for the SPC,S question type:

```
    !SPC,S,from position,length
```

SPC,T executes the Special block of questions immediately. See the SPECIAL compiler command in section *3.2.1 INTERVIEW CONTROL COMMANDS*. No location or width can be specified on the question label line.

Here is the syntax for the SPC,T question type:

```
    !SPC,T
```

SPC,U does a programmatic SUSPEND on the next question. See *3.2.1 INTERVIEW CONTROL COMMANDS* for more information on the SUSPEND compiler command. No location or width can be specified on the question label line. This option does not work in Practice mode. The interview will be resumed at the next question.

Here is the syntax for the SPC,U question type:

```
    !SPC,U
```

SPC,V stores the question number or label of the previously executed question. You can use this as the first question in a Terminate, Suspend, or Special block to quickly know at what point in the questionnaire the respondent went into this block;

the number or label stored is the question the interviewer was on when they typed the keyword or the question that caused the program to jump that block. In other parts of the questionnaire, it stores the last question that was "IF tested" (true or false) or asked, whichever is the most recent.

If the previously executed question has a label, the label is stored (whether or not there is a question number). If there is no label, the question number will be stored (whether or not there is a user-assigned question number). If the question number is stored, it will be in the format of QQ#.##. (For example, .5=QQ000.50, 1=QQ001.00, 23.5=QQ023.50)

This subtype requires exactly eight data columns. Specifying less will generate an error message. If for some reason you want to specify more than eight columns, control the width on the question label line.

Here is the syntax for the SPC,V question type:

```
!SPC,V
```

Adding ,2 after the !SPC,V gets the label of the question prior to the last question you executed. This is useful, for example, if the interviewer has suspended or terminated; the SPC,V,2 will get the label of the question they were at before they typed SUSPEND or TERMINATE.

SPC,W immediately writes out or updates a case rather than waiting until the end of the questionnaire; this is often used to save terminate data without using the standard 'go to the end' type of termination. By default, If the case ID is set with a SPC,A or SPC,D, you will get a case ID, otherwise you will get the ID of "NOTKNOWN".

Here is the syntax for the SPC,W question type:

```
!SPC,W<,type of data update to do>
```

The "type of data update to do" can be blank or "N" to write a new case or a second case with a duplicate case id, "U" to create a new record or update the current one (for data backup in case of a system crash), and "X" to generate a new case with a new case id, if you want respondent data from the same household or interview written in separate cases.

To write a single record, use !SPC,W or !SPC,W,N. This will not assign a case id unless one has already been specified using !SPC,A or !SPC,D. By default it assigns a case ID of "NOTKNOWNID". This is useful if you wish to write partial cases for screener terminates so that they don't use up a case id which would only be applied to completed cases.

To update or save partial data cases, use !SPC,W,U. Here's how this works:

If you specify an !SPC,W,U statement, and no case has been written yet, it will generate a newcase with a new ID (unless you have already assigned an ID using !SPC,A or !SPC,D. In this case, it uses that ID). Each subsequent !SPC,W,U statement updates the record already written.

And, at the end of the interview, there will be another update. This guarantees only one record per case id. This would most likely be used to save sensitive data in the event of a system crash, blow error, or other problem. You should check for and remove partial cases because this will lead to cases for interviews that were not completed.

To write additional cases within the same questionnaire, each with a new case id, you would use !SPC,W,X. This will write new cases each time specified with the "next" case ID assigned. You could for instance, write a separate case with a separate case id for each household member. The standard !SPC,W statement will also write multiple cases, but each with the case ID of "NOTKNOWNID" or the same case ID.

SPC,Y causes Survent to provide an answer for the next question it sees if an answer is not supplied in a certain number of seconds. This will fill any data-entry question type (CAT, FLD, NUM, VAR and TEX question).

Here is the syntax for the SPC,Y question type:

!SPC,Y,number of seconds to wait, response code to use

For example, on the initial phone status screen, so that if the interviewer gets to that screen and goes away, you can force an answer which will cause them to exit automatically out of the interview, so the supervisor and reports don't think they are still in the interview.

```
EX:
!SPC,Y,15, "No answer"
```

You could use this before a !VAR type data-entry question and it would be filled in after 15 seconds with "No answer" if not already responded to. The maximum response length for an !SPC,Y is 76 characters.

SPC,Z accumulates interview time. It works as a timer that you can start and stop.

Here is the syntax for the SPC,Z question type:

```
!SPC,Z,option
```

The options are:

0 start the timer or just record the time since the timer started

1 record time since the timer started and stops the timer

2 record time since last started

3 record time since last SPC,Z (any option).

Option 3 is used to record times between certain intervals, such as the time since the main questionnaire started, or the time since the questionnaire started and was completed, etc.

Here is an example of how it could be used:

```
{!spc,z,0} starts timer
"Screener questions are here"
{
!scrn: .3 !spc,z,3} - Time since start timer (Screener
time)
"Main questionnaire questions"

{main: .3 !spc,z,3} "Time since scrn "main" time
{Tot: .3 !spc,z,2} Total time
```

The time accumulates in seconds. If a data location is not specified on the question label line, the time is stored internally (and, when suspended, in the Suspend file). The time is stored as an integer (whole number), zero-filled. If the field specified is not large enough, it is filled with asterisks.

The typical scenario would be to start the timer at the point that you want to start counting, and record it with SPC,Z,2 at critical points in the questionnaire (i.e., when suspended). Stop it at the end of the interview, and record total time with SPC,Z,1, assigning a data location.

This subtype gives you complete control over what parts of the interview are timed.

You don't have to turn off (stop) the timer during suspends; it knows not to count time when suspended.

SYSTEM Statement

SYSTEM statements are used to access outside systems. !SYS,1-!SYS,3 allow you to escape from the Survent program to execute outside programs, use operating system commands, or talk to the modem port in DOS or UNIX. !SYS,L lets you change languages, and !SYS,Q lets you quit the interviewing session.

!SYS,1 executes an external program. When the SYS,1 executes, Survent is exited, and saves the existing data case as "cfmd####" where "####" is the station number of the device executing the SYS command. Once the external program has finished, the file cfmd#### is read back in to the memory.

You must have enough memory on your computer to run Survent and the other program simultaneously.

```
EX:
{dolotus:
EDIT MYFILE
LOTUS %1 %2
!SYS,1 }
```

!SYS,2 sends operating system commands. When the SYS,2 executes, Survent is exited, and saves the existing data case as "cfmd####" where "####" is the station number of the device executing the SYS command. Once the external program has finished, the file cfmd#### is read back in to the memory.

```
EX: {
cp /u/cfmc/fred*
dir fred*
!SYS,2 }
```

!SYS,3 talks to a port on the local computer. Here is the syntax and example for the !SYS,3 statement type:

```
{label:
<command string 1>
<command string 2>
<command string n>
!SYS,3,<port #>,<baud rate>

EX:
{
!SYS,3,2,2400}
```

This initializes a modem on port 2 to 2400 baud.

```
{
atdt14157770470
!SYS,3
```

This sends an “AT” command string to the modem. Once the port and the baud rate have been identified, they do not need to be re-specified.

!SYS,I sends data to the end of the INTVRLOG record for this interview. The statement is designed so that you can add custom information to the standard cfmc log records saved in the INTVRLOG file. The syntax is:

Syntax:

```
{
<standard CfMC text line with \: & \| stuff>
!sys,i,<column in log record>,<width> }
```

This will put the specified text line into the INTVRLOG in the user area (2001-3000) of that interviewer's log record. The column must be between 2001 and 3000, and the column+width must be <=1000.

You can also retrieve what you have stored in the interviewer log file or other information in the file by using the syntax "\|%column.width|".

!SYS,L sets the language to be used in the questionnaire from this point. Languages must have been programmed into the questionnaire for this to take effect. Interviewers may also set the language by using the “L=xx” command where “xx” is the language they wish to use.

```
EX:
{!IF Lang(F)
!SYS,L,FR }
```

If you are using “1 character language coding” by specifying “!language set=(e=en f=fr)”, then on the !SYS,L command you also use one-character codes such as “!SYS,L,E”.

!SYS,Q,QUIT/LUNCH/BREAK Causes the interviewing session to end immediately after the current interview. It will save the current datacase if the interview is completed. If an !AFTER_QUIT block is used in the questionnaire, it will execute the questions in that block after ending the interview and before ending the interviewing session, and it will generate a separate datacase with the same case ID for the answers in that block.

!SYS,Q,QUIT/LUNCH/BREAK allows you to specify the reason for the exit from the interview. For instance, if you enter "!SYS,Q,LUNCH" to send an interviewer to lunch, it will display a note on their terminal that they have gone to lunch.

!SYS,T,<timezone> allows you to specify which time zone the interviewer is in and pass it to the CfMC server so that calls get routed properly, according to which time zone the interviewer is in. The time zone can be either a three-digit or two-digit number. If it is two digits, it is a regular CfMC time zone 01-24. If it is three digits the last digit is "5" then it is a "half" time zone, eg. "225" would be time zone 22 and 1/2.

This feature is designed for webCATI applications where you cannot predetermine which time zone an interviewer will be in. In other cases you can set the time zone as part of the login script or in the CfMC ttyinfo file. But, it will work in either terminal mode or Web mode.

3.1.6 Call Questionnaires

Call questionnaires allow you to create sub-questionnaires that are called by the main questionnaire. These sub-questionnaires can be called as many times as needed on a given survey.

For instance, if you're asking people about banks they use, you could create an overall demographic questionnaire and have it call a questionnaire for each person in the household and then have this questionnaire call another questionnaire for each bank that this person might use.

There is no effective limit to the number of call questionnaires or levels. CfMC has tested 150 sub-questionnaires five levels deep in one study.

You must have an SPC,A statement before the first CALL question so that a case ID is assigned for the respondent. This is necessary to tie all the child interviews to the original parent.

Use the local scratch area (SPC,K and L) to pass data from the parent to/from any child. Your local scratch area will retain information across interviews within a called set of questionnaires. !CALL statements can be placed within !RESUME, !SUSPEND, !SPECIAL and !AFTER_QUIT blocks. Here is the syntax for call questionnaires:

```
Syntax:
!CALL,subtype,<questionnaire to call>
```

Subtype 1 is the default. It redirects the interview to another QFF file, but uses the original FON, QUO, etc. files. A separate data record is saved for each called questionnaire executed.

When the called questionnaire is completed, it returns to the point it where was called in the original questionnaire.

Subtype 2 allows changing of phone, quota, and data information. It redirects the interview to another QFF file, and gives total control to that QFF file. The QFF file, FON file, QUO file, etc., all use the new study name. It lets you keep information in the local scratch area as you switch from one questionnaire to the other (e.g., carry respondent's name from the main to the called questionnaire). This would be used for example, if you want a menu of studies for an interviewer to choose from.

Subtype 3 can be used instead of !CALL,1 if you have a case where you don't want the calledquestionnaire loaded when the calling questionnaire is loaded, but only when specifically requested. When the called questionnaire is completed, it returns to the point it where was called in the original questionnaire.

Subtype 4 can be used in place of !CALL,2 if you do not want the called questionnaire loaded when the calling questionnaire is loaded, but only when specifically requested.

Use subtypes 3 or 4 if you have a dynamic list of possible questionnaires to run, and you don't want interviewers running them on certain days. Or the questionnaires may not be available at that time. Therefore, you won't get an error if you load the calling questionnaire and the called questionnaire does not exist.

In conjunction with subtypes 3 or 4, you can use a special conditional statement to check for the existence of a questionnaire before loading it:

```
Syntax:
!IF xf(usable_study( <qffname> )
```

NOTE: You can use Coding mode on a CALLed questionnaire.

WARNING: When specifying !Call,2 or !Call,4, the current data and quota file are NOT updated unless you use !SPC,W to update the data and all quota statements use the !quota,,name,+#,NOW option for immediate update. This operates like the !SPC,B statement to abort an interview. So, when returning to the main menu via a !Call,2 or !Call,4 statement, be sure to specify the !SPC,W and !QUOTA,...,NOW statements first.

If you “abort” or use !SPC,B or !SPC,H to abort a called interview, it returns control to the calling interview if using !CALL,1 or !CALL,3, or returns to the interviewing prompt using !CALL,2 or !CALL,4.

The example specification below creates different QFF files: CALL and DEPT. CALL calls DEPT. This example limits DEPT's number of executions by the answer you give to a NUMERIC question. See \CFMC\Survent\INDEX for more examples.

Two sets of files are created, including two TR files (thus, the need to assign the case ID).

```
~PREPARE COMPILE
[CALL]
{!SPC,A } ''assign case ID now
{DMENTS:
How many departments in your company?
!NUM,,,1-5,,DK }
{!IF DMENTS$="DK"
!GOTO,ALLDONE }
{KEEPTRAK:
!EXP,,0 } ''initialize counter
{COMEBACK: [KEEPTRAK] ''counter to keep track of how many
                        times
!EXP,,[KEEPTRAK]+1 } ''you've gone to the Call qstnr.
{GETSURV:
!IF [COMEBACK]<=DMENTS ''if counter <= # depts., continue
                        and
!CALL,,DEPT } ''call the DEPT questionnaire now
{
Welcome back! ''this is where CALLed questionnaire
                  returns
Press <ENTER> to continue.
!DISP }
{!IF [COMEBACK]<DMENTS ''if counter < # depts., go back
                        to CALL qfile
!GOTO, COMEBACK } ''until have hit maximum.
{ALLDONE:
!GOTO }
~COMMENT
~PREPARE COMPILE
[DEPT]
(Department questionnaire specifications here)
~END
```

Call,1 Questionnaires and Quota Files

If you are using !Call,1 or !Call,3, and you will be updating quotas in the child questionnaire, then the quota structure for each child must match the parent. Do this by having a block of matching quota references (like !TARGET commands) at the top of each questionnaire.

Call,1 Questionnaires with Suspend Blocks

Using !CALL,1 or !CALL,3 questionnaires, the last question in the Suspend block *must* be executed. The simplest way to guarantee this is to use a passive (receiving) GOTO question as the last question.

3.2 QUESTIONNAIRE CONTROL USING COMPILER COMMANDS

PREPARE's compiler commands control questionnaire functions. Depending on their placement, they may control the whole questionnaire or just a portion of the questionnaire. Compiler commands may be specified individually or in groups.

Many of these commands are like switches. They are turned on to perform a function, and they must be turned off afterwards. Some of them start and end command blocks (for example, a ROTATE and END_ROTATE pair).

All compiler commands start with {! and end with }; you can specify an optional label if you wish, using this syntax:

```
{ label: !command }
```

Using a label for a command lets you skip to the command instead of having to put a dummy GOTO just before it.

NOTE: All commands may be commented out by placing a minus sign (-) before the label or "!".

Some compiler commands may be turned off by placing a minus sign (-) immediately after the exclamation sign. A reminder of this feature will appear with the specific command.

There are four types of compiler commands. They are:

1. Interview controls, which control operations during the interview, such as rotation of questions, keyword operation, and screen format.
2. Compose controls, which control the composing of questions.
3. Compile controls, which control compile operations and the format of the files made at compile time (hardcopy, backup).
4. Data controls, which affect the placement of the data in the data file(s). These are quite useful when using column-free data specifications.

Compiler commands must be placed after the study header and between question specifications and other compiler commands.

3.2.1 Interview Control Commands

{! _only} causes interviewer to only get numbers in specified markets from !PHONE,L.

This command forces the interviewer to only get records from the specified market if you are using !PHONE,L to get numbers from a certain market. By default, if there are special interviewer numbers or timed numbers that are in a different market, the program will get those numbers for the interviewer, who must decide what to do with the numbers.

But if GET_MARKET_ONLY is specified, a number is picked up from another market, the program puts it into "bucket 9" for the market it picked the number from. If you want these numbers delivered to someone else working in that market, make sure you use the "Release_bucket_9=4" option to release those numbers and override the "minimum system callback time" for the number and call it immediately.

You can turn this off with {!-GET_MARKET_ONLY}.

{!AFTER_QUIT} {!END_AFTER_QUIT}

Commands in this command block will be executed after the interviewer types “Quit” to end their interviewing session, or if the program forces them to quit by using a “!SYS,Q” command. This is typically used to gather summary information on the interviews or offer the interviewer options when they are quitting the session. It will create a separate data record from the one that is stored for the interview and it will generate its own case id. !GRID, !LOOP and !CALL statements can be used within this block. However, you cannot !ROTATE a block within this block.

{!ALLOW_ABORT}

Allows the interview command ABORT. -ALLOW_ABORT is the default (unless ALLOW_ABORT is specified in the header statement) and need only be specified to override a previous ALLOW_ABORT.

{!ALLOW_BACKUP}

Re-enables backing up in the interview using the caret (^) after this point. This is the default. -ALLOW_BACKUP will disable ^ until the end of the interview or an ALLOW_BACKUP is specified.

{!ALLOW_LANGUAGE_CHANGE}

Controls whether the interviewer can change languages by typing "L=". The default is "no', that the interviewer cannot do this unless they are in "debug" mode.

{!ALLOW_MONITOR}

Allows monitoring of the following part of the questionnaire by SURVSUPR or SURVMON. This is the default and need only be specified after a -ALLOW_MONITOR has disabled monitoring.

{!ALLOW_RESET}

Re-enables the Survent interview command RESET. The default is *not* to allow RETAKE. -ALLOW_RESET will disable RESET until the end of the interview or an ALLOW_RESET is specified.

{!ALLOW_RETAKE}

Re-enables the Survent interview command RETAKE. The default is *not* to allow RETAKE. -ALLOW_RETAKE will disable RETAKE until the end of the interview or an ALLOW_RETAKE is specified.

{!ALLOW_SPECIAL}

Allows the interviewer to use the keyword SPECIAL. This is the default and need only be specified after -ALLOW_SPECIAL has disallowed the use of the keyword.

{!ALLOW_SUSPEND}

Re-enables the Survent interview command SUSPEND. This is the default. -ALLOW_SUSPEND will disable SUSPEND until the end of the interview or an ALLOW_SUSPEND

is specified.

{!ALLOW_TERMINATE}

Re-enables the Survent interview command TERMINATE. This is the default. -ALLOW_TERMINATE will disable TERMINATE until the end of the interview or an ALLOW_TERMINATE is specified.

{!ALLOW_TEXT_EDIT}

Allows the interviewer to modify a TEX question response in the on-line editor. This is the default. -ALLOW_TEXT_EDIT will force the interviewer to re-enter the response either for an EDIT question or when backing up to the question. (UNIX line mode only).

{!AUTO_RETURN}

Causes responses to be accepted as soon as they reach their maximum length without the need for an **Enter**. An **Enter** will still be needed for responses having less than the maximum number of characters or responses.

For single-response CAT or FLD questions, this would be as soon as the code is typed. For multiple-response CAT or FLD questions, this is when the maximum number of responses is typed.

For NUM or VAR questions, this is as soon as the maximum length of the question is reached. This is especially useful for keypunch type applications, where speed is important, or for self-administered questionnaires where you want to move directly from one screen to the next.

This can be turned off during a session with -AUTO_RETURN.

AUTO_RETURN overrides HIGHLIGHT_CATS and ECHO_CATS. Enter the response code of the desired choice and it will accept and move on immediately, without an additional **Enter**. You may also arrow through the list (on monitors; use **Ctrl**-**U**,**D**,**R**,**L** for terminals) to an item and then choose it with **Enter**, or enter a response code for a different item. For multiple-response questions, you'll need to arrow to the appropriate code, then press **INS** or a plus sign (+) to accept. See HIGHLIGHT_CATS below. There will be no automatic **Enter** generated; you'll need to press **Enter** when you have all the appropriate responses highlighted.

{!BACKUP_HERE}

This option controls where you will back up when using the BACKUP key, usually "^" but also whatever key is specified on the "BACKUP: xx" parmfile command. By default you back up to the previous question asked, but if you specify BACKUP_HERE before a question and back up from a question further down in the questionnaire, it will back all the way back to the question asked just after the BACKUPHERE command. To return to regular backup mode, use the BACKUP_WHEREAT command.

{!BACKUP_WHEREAT}

See the BACKUP_HERE command above. This returns you to regular backup mode.

{!BEFORE_SESSION} block executes as first set of questions in an interview session. This will be executed before the first interview of any questionnaire, similar to the other special blocks. The syntax is:

Syntax:

```
{!before_session}
<questions>
{!end_before_session}
```

Use this to indoctrinate the interviewer or start counters. This can be used instead of using a flag in the local scratch area to only display a set of questions once.

{!DISPLAY_MODE=terminal/webcati/websuper/

websurvent} controls what things are allowed in compiles

This new keyword now controls what things are allowed in a compiled questionnaire. !DISPLAY_MOD=WEBSURVENT replaces !WEBSURVENT as well. This is used to make sure you don't use features that were designed for some other display mode. You will get errors, for instance, if you try to use a !GRID,R in webSurvent mode.

{!ECHO_CATS}

Places the interviewer in Echocats mode. -ECHO_CATS will turn it off. See the header option ECHO_CATS for more information.

{!ERROR_LEADIN text }

This command allows overriding the "ERROR #" leading text on error messages; typically this is to use some other language for the word "ERROR". Note also that the text "ERROR: " is in msgfile message #7222 and can be given other language equivalents there.

{!ERROR_MSG #### text }

Sets the value of the numbered message #### in the CfMC MSGFILE to “text”. This is so you can replace standard CfMC error messages with your own error messages or warnings for a particular study without rebuilding the MSGFILE.

You can use this command more than once for the same message number if you want to change the message during the survey, for example:

```
     EX:
{!Error_msg 7611 Please enter a valid value from 1 to 10 }
... (Questions where this messages could occur)
{!Error_msg 7611 Please enter a valid value }
```

If there are multiple languages in your questionnaire, you can include changed error messages for all languages using the syntax:

```
     Syntax:
     {!error_msg NNNN
     xxNNNN <xx language message for NNNN>
     yyNNNN <yy language message for NNNN>
     NNNN default message for NNNN
     }
```

Where "xx" is the language and "NNNN" is the message number.

You may use “back-references” to display prior responses in the message like you can with any other CfMC question text, for instance, this adds the response to Q1 to the message:

```
     EX:
{!error_msg 1234 response does not match previous (\:Q1:) }
```

{!FIELD_SAMEAS_USE_ORIGINAL_WIDTH}

This directive controls what width to use when using the "[SAMEAS label]" feature on response lists. This has been added to allow users to set a default width for questions using the same list, in the case where you want to pre-allocate columns that may be used for coding later.

The default is to generate a width based on the standard formula for !FIELD quesitons, which is "(# codes allowed) * (code width)". If the keyword above is specified, the program will use the width of the originating question instead; so if you have a width specified on the original question, or you allow more codes on the originating question, it will use the width generated for that question when assigning widths for the succeeding "[SAMEAS label]" using questions. For example:

```
EX:
{!fld_sameas_use_original_width}
{Fld1:
!fld,,5
01 code 1
etc. }
{
Fld2:
!fld
[sameas fld1] }
```

In this case, "Fld2" will have a width of 10 columns, because that was the width of Fld1. If the keyword was not there, Fld2 would have a width of 2 columns.

{!FILL_VIEW}

"VIEW" mode by default shows the responses to questions as they are executed in the view session. This can cause some logic problems. For instance, if you have quota checking statements and the quotas have changed since the questionnaire was recorded, it could cause the logic to change.

Fill_View says "Fill the answers to questions from the data regardless of the calculation in the VIEW session or whether the question is responded to". In this way, HIDDEN questions and EXPRESSIONS will be filled exactly as they were filled in the original data. Note that this may be a bit misleading within the VIEW session as some response will be displayed as if they were

answered if queried when they were just used as a placeholder variable. This does not change the data that was actually recorded at the time of the interview, it is just a matter of how the VIEW session executes.

{!FLAG_DISALLOWED_CATS}

Turns on the flagging of categories that are not allowed on CAT or FLD question lists (typically when using subtypes I, C, or D). Turn back off again by using -FLAG_DISALLOWED_CATS.

{!HELP_ON_TOP}

Puts help lines generated by the !HELP_type compiler directives or the !HELP command on a question on the top of the screen, instead of at the bottom, which is the default.

{!HELP_type text }

Puts a line of help at the bottom of the screen for the question type specified. If no text is specified, a default message is displayed. If !HELP is specified on a particular question, it overrides the !HELP_type message. If the !HELP_ON_TOP is specified, the help message goes to the top line of the screen instead of the bottom. See *3.2.5 HELP COMPILER COMMANDS* for more information.

{!HIGHLIGHT_CATS option(s) }

Creates a special screen format for category (CAT) and field (FLD) type questions. The user is positioned on the first category, which is highlighted. For single-response questions, to accept the category, the user presses an **ESC** or **Enter** (unless = was used as the first code; interviewer must first move to a valid response). To move to another category, the user types the response code, or uses the arrow keys (on a monitor; use **Ctrl-U**,**D**,**R**,**L** on a terminal) to move to the relevant category.

For multiple-response questions, the space bar or "+" key is used to accept a response, and the space bar or "-" key takes away an accepted response (monitors can use the **INS** and **DEL** keys; terminals can use **Ctrl-N** in addition to the "+" key and **Ctrl-P** in addition to the "-" key). **ESC** or **Enter** accepts the question after all responses have been marked. This is especially useful for self-administered questionnaires.

This can be turned off during a session with -HIGHLIGHT_CATS.

You can specify various options after !HIGHLIGHT_CATS to control what type of questions to use it on and what to do when a response is specified.

The options ALL, SINGLE, and MULTIPLE, control which types of questions to apply HIGHLIGHTCATS to. ALL is the default, SINGLE affect single response questions only, and MULTIPLE affects multiple response questions only.

The options ERASE and SELECT control what to do when an item is picked. “SELECT” puts highlightcats in SELECT mode on whatever question type is specified. “SELECT” mode is the standard mode for multi-reponse lists, where you must select a code using the spacebar or “+” key or INS key to go on. Specifying {!HIGHLIGHTCATS ALL SELECT} forces single response questions to be treated the same as multiple response questions.

The option ERASE causes the text for the response to be ERASED when chosen. It can also be used with the ALL, SINGLE, and MULTIPLE statements.

{!KEEP_RESPONSE_CODES}

This displays the response code in addition to the text of the response when using \#(label) or \:label: to display the answer to a response list item. For instance, it will display something like "01 Likes Apples" instead of just "Likes apples" when back-reference a question which was answered with “01”.

{!LANGUAGE SET=(lang1 lang2 langn) CHARACTER_SET=charset SPEAKING=langx }

The compiler directive {!LANGUAGE ...} controls what languages you are using and which character set. "SET=" IS required. The languages being used can have 2 character codes or can have the 2 character codes converted to a 1 character code for ease of coding, for instance:

```
EX:
{!LANGUAGE SET=(e=en f=fr s=sp) }
```

This says to use English, French, and Spanish, and allow coding of them as 1 character codes in the questionnaire (eg. \le, \lf, \ls).

The "character sets" are ASCII, Extended_ASCII, Shift_jis, Multi_byte, or Universal, the default is “Universal” which allows all character sets.

The "Speaking=<language>" parameter would be used if you want to set the starting and default language; if you don't specify it it defaults to the first language on the list.

{!MAXIMUM_CONSECUTIVE_BACKUPS=#}

This command controls the number of consecutive backup commands that can be issued.

Backup commands include “^”, “reset”, and “retake.” (See *4.3 SURVENT COMMANDS* for more information.)

You can enter {!MAX_CONSEC_BACKUPS=1} to only allow users to back up one question at a time without going forward again. Since users have to retype data (or use the "=" command to save the answer) after backing up multiple questions, this will keep them from mistakenly losing their answers.

In addition, commands “!Backuphere” and “!Backupwhereat” allow you to control where interviewers back up to if necessary when they enter “^” to back up. This allows you to force respondents to start at the top of a section when they back into the section. Intermediate questions are cleared.

{!READ_NAMED_QUOTAS}

This does an immediate read of the quota file as it now stands. This eliminates the need for many QUOTANOW functions. This can be done anywhere in the questionnaire, but is usually used at the point in the questionnaire that you are doing quota checks. Any subsequent QUOTA functions will reflect the values at the time of the READ.

{!RESPONSE_LIST_COLUMNS=#}

This allows you to have an even number of codes in each column for as many columns desired on a screen (as long as the items will all fit). This lets you have a more standardized format for your lists as they are displayed on the screen.

If you are using Highlightcats mode, the program will split the codes into columns across pages.

If you use "1" as the number of columns, you will force Survent to put each item on a separate line.

The maximum number of columns you can specify is 4. If Survent cannot display the list in the number of specified columns, it will reduce the number of columns to make it fit.

{!RESUME}

{!END_RESUME}

These commands mark a block of questions for execution when an interview is resumed. Questions are otherwise ignored. You cannot branch into or out of (GOTO, SKIPTO) a Resume block. !GRID, !LOOP and !CALL statements can be used within this block. Interviewers may not type SPECIAL, SUSPEND, or TERMINATE while in this block. Note that you cannot !ROTATE blocks inside these blocks.

{!RESUME_HERE}

Interviews suspended after this statement will come back here to start. Often used at thebeginning of a block of related questions, such as before a Rotate block. Use as many in a questionnaire as needed.

{!RESUME_WHERE_AT}

Used after a RESUME_HERE to say “resume at the question suspended at” from now on. This is the default and need only be specified to turn off a previous RESUME_HERE. Use as many in a questionnaire as needed.

{!ROTATE,type,#,label}

Begins a Rotate block, causing the questions to be asked in some type of rotating order (for specific information on types and uses, see *ROTATING QUESTIONS* following this section).

{!END_ROTATE}

Ends the Rotate block. This is placed at the end of the questions to be rotated. Required with ROTATE.

{!FIX,n}

Within a Rotate block, this fixes the position of the next *n* questions. This must follow a ROTATE command. If *n* is specified, that number of questions cannot include other rotate commands. If *n* is omitted, all questions until the next relevant command (FIX, GROUP, or END_ROTATE) at the same rotate level are fixed. You cannot fix the first question inside a Rotate block. (For more information see the *FIX AND GROUP* section immediately following *ROTATING QUESTIONS*.)

{!GROUP,n}

Within a Rotate block, this keeps the next *n* questions together as a group when they are rotated. This must follow a ROTATE command. If *n* is specified, that number of questions cannot include other rotate commands. If *n* is omitted, all questions until the next relevant command (FIX, GROUP, or END_ROTATE) at the same rotate level are grouped. (For more information, see the *FIX AND GROUP* section immediately following *ROTATING QUESTIONS*.)

{!RECODE_ROTATE_ORDER}

This command captures the order of response items displayed. It has the same format and meaning as the !ROTATE_ORDER command, except that it is designed to capture the order of items on a !FIELD or !CATEGORY rotate list. This command supports both !FIELD/!CAT subtype "R" rotations, or using "!Rotate" on the response list. The syntax is:

```
SYNTAX:
{: .10 hide !var }
{!Recode_rotate_order,[,noblow] }
```

You first need a question to store the result in. This needs to be long enough to store the order, for instance, if you have 101 codes, it needs to be 3 * 101 = 303 columns long to store the order for all questions.

Then the statement just requires "!Recode_rotate_order,label". It will store the position of the questions in the order they are asked, for instance, "4321" if the 4th code was displayed first, the 3rd code displayed second, the 2nd code displayed third, and the 1st code displayed last.

Note that if you have 5 !ROTATE blocks but only one !Recode_rotate_order statement, each !Rotate block in succession will save its order to the question referenced in the !Recode_rotate_order statement unless you use {!-Recode_rotate_order} to turn it off.

Sometimes you will only want to record the first "N" items asked; to do so, set the variable length to the length need for that many items and add the parameter "noblow" to the statement. This will allow the program to save just that many items without blowing due to lack of space.

{!ROTATE_ORDER,label}

The "ROTATE_ORDER" compiler directive stores the order of the questions asked in a following !ROTATE block. It will work for all rotate types, but is most useful for !ROTATE,S (scramble rotate) as there was no easy way to record the order. The syntax is:

```
SYNTAX:
{<label>: <location>.<length> hide !var }
```

{!ROTATE_ORDER,<label>[,asked,noblow] }

You first need a question to store the result in. This needs to be long enough to store the order, for instance, if you have 101 questions, it needs to be 3 * 101 = 303 columns long to store the order for all questions.

Then the "Rotate_order" statement just requires "!Rotate_order,label". It will store the position of the questions in the order they are asked, for instance, "4321" if the 4th question was asked first, the 3rd question asked second, the 2nd question asked third, and the 1st question asked last.

Note that if you have 5 !ROTATE blocks but only one !Rotate_order statement, each !Rotate block in succession will save its order to the question referenced in the !Rotate_order statement unless you use {!-Rotate_order} to turn off the storing of rotate orders.

To limit the list to the items that are askable, eg. to exclude !generate or !expression statements in the rotate, add the parameter "asked".

To record only the first "N" items asked, set the variable length to the length needed for that many items and add the parameter "noblow" to the statement. This will allow the program to save just that many items without blowing due to lack of space.

{!SCROLL_SCREEN}

This forces sections of the questionnaire to be in “SCROLL” mode by default (see \A text controller) instead of being in “CLEAR SCREEN” mode (default); this means that each question will be displayed below any other questions on the screen instead of clearing the screen first. If you have the “\T” directive to clear the screen, or use a screen positioning command “eg. \(5,10)”, it will override the default for that question.

{!SET_INTERVIEWER_CAPABILITIES=+-options}

This allows you to change interviewer capabilities on the fly. Be careful when toggling "training mode" on and off because it will cause you to lose data if you turn it on during a real interview. But this allows you to make a menu of options instead of using the keywords at the survent command prompt.

The possible options are:

B can Abort or Break or Change interviews

C can Change interviews

D DEBUG mode (overrides interviewer controls, allows GOTO forwards and backwards, can Abort, Break, or Change interviews)

N Interviewer has no capabilities, and cannot VIEW prior interviews

E interviewer started in Echocats mode

L do interviewer logging and save at end of interview

Q do interviewer logging and save after every question

T interviewer started in Training mode

```
     SYNTAX:
{!SET_INTERVIEWER_CAPABILITIES=BCDENTLQ}
```

For more information on these options, refer to !SPC,5, which describes each of these in more detail.

Using “+” before the list of letters means the capability will be in addition to whatever is already set, using “-“ means to remove that capability from the interviewer.

{!SHOW_LAST_RESPONSE}

This allows you to show what was entered on the previous question at the bottom of the current screen, if space is available. This is useful so that users don't have to back up to remember what the answer was to the previous question. This, of course, will only show the very last response and not the contents of grids or within a grid. If more than one line of information was entered, the line is truncated. {-SHOW_LAST_RESPONSE} (in its negative form) turns off the directive.

{!SHOW_QUESTION_LABELS=option}

Displays question labels in the corner of the interviewer's screen. The default is to not show them. Options are UR to show in the upper right corner, or LR to show in the lower right corner.

{!SPECIAL}

{!END_SPECIAL}

These compiler commands mark a block of questions for execution when an interviewer enters the keyword SPECIAL. This can be done at any place in the questionnaire except in Suspend or Resume blocks, and can be done more than once during an interview, so put some logic in the Special block to control what happens if you enter the block more than once. At the end of the block the interviewer will return to the point in the questionnaire where she/he had entered SPECIAL.

You may put this block anywhere in the questionnaire; in the normal flow of the interview, the program will branch around this block without you having to tell it to. You can programmatically execute the Special block by using an SPC,T. You may not skip to a question inside a Special block, or skip out of one. !GRID, !LOOP and !CALL statements can be used within this block. Interviewers may not use commands SPECIAL, SUSPEND, or TERMINATE while in the Special block. They may, however, use the caret (^) to back up within a Special block (and possibly back up out of it as well or, having finished the block, back up into it again). Also, note that you cannot !ROTATE a block within this block.

{!SUSPEND} {!END_SUSPEND}

These commands mark a block of questions for execution when an interviewer enters the keyword SUSPEND. Questions are otherwise ignored. You cannot branch into or out of (GOTO, SKIPTO) a Suspend block, but you can back out of one by using the caret (^). !GRID, !LOOP and !CALL statements can be used within this block. You can programmatically execute the Suspend block by using an SPC,U. Interviewers cannot use keywords SPECIAL, SUSPEND, or TERMINATE while in this block. Also, note that you cannot !ROTATE a block within this block.

{!VIEW}

{!VIEWX}

These commands will turn on or off the allowing of View mode for a section of the questionnaire. !-VIEW will turn off View mode for a particular section of the questionnaire until a !VIEW or !VIEWX is seen. Use this to cause quota checks and choose procedures based on quotas not to be re-executed during the viewed interview.

!-VIEWX will not show questions in that section but will execute them. Use !-VIEWX when you don't want to see a block of questions in View mode, but still need them executed (skip patterns, for example), so that the interview will behave the same as the original.

{!VIEW_QUOTA=letter}

This controls how Survent will evaluate QUOTA functions read while in View mode. The default is that IF conditions with quotas on them will be false and EXP questions will return a “0” for the quota value. This is to avoid complications trying to execute the questionnaire using current quota values when the data was collected with quotas that probably were different.

VIEW_QUOTA=E means “evaluate”, and it will return whatever the current value is for the quota reference in the quota file.

VIEW_QUOTA=F (false for IFs) and VIEW_QUOTA=N (O in EXPs) are the defaults.

VIEW_QUOTA=T will return True for IF conditions and EXP QUOTA ( ) references will return 1.

VIEW_QUOTA is overridden by -VIEW; if -VIEW is set, the whole section will not be executed.

VIEWX after -VIEW will return control to the VIEW_QUOTA.

ROTATING QUESTIONS

Using the rotate compiler commands, you can create blocks of questions that will be presented to the interviewer in a different order for each interview. You may choose from three types of rotates. You may also rotate some or all of the questions in a block, and may have a Rotate block within another Rotate block. There is a default limit of 10,000 questions within the outermost level of the rotate. This includes all compiler commands, other rotate commands, etc. To increase the limit, use the header option MAXIMUM_QUESTION_SIZE.

A Rotate block must start with a ROTATE compiler statement and end with an END_ROTATE compiler statement. The syntax for a ROTATE compiler statement is:

```
{!ROTATE,type,# per group,controlling label
or data location}
```

The rotate type, number in group and controlling label are all optional. If you don't specify a type, the default is R (random start). The rotate type specifies how the questions will be rotated:

R A random start rotate starts at a random question in the block (per interviewer, per session), and asks the questions sequentially from that point until all the questions in the block have been asked. In the next interview, with the same interviewer in the same session, the first question asked will be the next one of the rotate series (e.g., 1st time: 3,4,1,2; next time: 4,1,2,3; next: 1,2,3,4, etc.).

S A scramble rotate asks the questions in random order each time. It is possible, but unlikely, that questions will be asked in the same order twice in a row.

F A flip rotate asks the questions in either the original order or reverse order of the rotate series. The first time through may start at either the top or the bottom of the set of questions. The second time around the questions are asked in the opposite order. The third time around the questions are asked in the first order, etc. (e.g., 1st time: 1,2,3; next time: 3,2,1; next: 1,2,3).

If you have a standard number of questions to keep together when rotating, you may specify this.This number must divide evenly into the total number of questions rotated. You may, for example, have ten questions in a rotate block and want every two to be rotated together.

Specifying a number per group is optional; the default number is 1. See the GROUP command for more flexibility in keeping questions together.

Rotates can be controlled by a number in the data. For instance, you could direct the program where to start the rotate for a regular rotate. This is specified by using a GEN, NUM, or EXP statement to generate a number, and referring to that question's label or location on the ROTATE compiler statement. You can also control rotates by specifying a label for the ROTATE compiler statement itself and then referring to it as the controlling label for other rotate statements. In this way you will be using the number that the program puts in the data rather than generating one yourself with one of the statements mentioned earlier in this paragraph.

Specifying a controlling label in a ROTATE compiler statement is optional. If you specify a blank data location, PREPARE will store the rotate information in that field. The field used will be stored in the DB file as a variable that you can run reports off of.

It is good practice to put a RESUME_HERE compiler command before the ROTATE and a RESUME_WHERE_AT after the END_ROTATE.

You can't set a rotate seed for an internal rotate once you are inside the external rotate; when the external rotate is read, rotate controllers are set within the rotate, and changes in the data will not change the rotate. All controllers must be set prior to the execution of the outermost rotate.

ROTATE compiler statements use data columns. A warning message is given if any hardcoded data columns are going to be overwritten. Here are the number of columns used by each, and what goes into those columns:

Rotate type	**What goes in data**

R	A three-column field holds the number of the question in the block that the rotate will start with (this will be a number in the range of 1 to the number of questions).
S	A six-column field holds the number which is a "seed number" that is used by the program to randomly assign the order.
F	A one-column field holds 1 to go forward or 2 to go backward.

TIP: For resumed interviews, put a {!RESUME_HERE} statement before each {!ROTATE}, and a {!RESUME_WHERE_AT} after each {!END_ROTATE} statement. This guarantees that all of the questions in a given Rotate block will be executed when the interview is resumed.

Controlling Rotate Sequence During an Interview

This clarifies how the rotate sequence is controlled for Rotate blocks during an interview:

If you have multiple Random start Rotate blocks then each block will get a different starting point. For a Flip rotate, the flip sequence gets its seed from the question number of the last question in the block; if odd, it rotates back to front, so even will rotate front to back in the same interview.

Unless you are assigning question numbers with this in mind, you must control the rotate sequences with a controller and then add a number to the controller to change the sequence for a particular block.

Scramble gets its seed from the system clock. Using a controlling question, you can have the program control two rotates in exactly the same manner. If you want the program to decide the sequence, but want two or more sections to rotate in the same manner, assign a label to the first rotate statement then refer to it as the controlling label in subsequent rotates. The program places a number that controls the rotate in the data. If you specify this same location or label as the controller for another rotate, the program will read the code from the specified data and treat the second rotate the same.

The syntax for an END_ROTATE compiler statement is:

```
     {!END_ROTATE}
```

NOTE: If you choose to, you can label an END_ROTATE, so it is possible to GOTO it.

```
     {label: !END_ROTATE}
```

Here are two sample Rotate blocks:

```
     EX:
     {!ROTATE,R}
     {FIRSTQUE:

     Do you like apples?
     !CAT
     1 YES
     2 NO}
     .
     .
     {LASTQUE:
     Do you like oranges?
     !CAT
     [SAMEAS FIRSTQUE]}
     {!END_ROTATE}
```

In this block note that the specified rotate type is R — this is a random start rotate. No other options are specified—there is not a specified number of questions to be rotated together, and there is no controller question. The program will start at some question and ask all questions sequentially to the end of the block, then return to the beginning of the block and ask the questions up to the first one it asked.

```
     EX:
     {!ROTATE,S,3,USES}
     {FIRSTQUE:
     Do you like apples?
     !CAT
     1 YES
     2 NO}
     .
     .
```

```
.
{LASTQUES:
Do you like oranges?
!CAT
[SAMEAS FIRSTQUE]}
{!END_ROTATE}
```

In this block note that the specified rotate type is S—this is a scramble rotate. Every three questions will be rotated together, so the number of questions in the rotate block must be divisible by three (6, 9, 12, . . . ,30, etc.). The controller question is USES, so the rotate will be controlled by the number in that question. USES can be the location of a NUM or EXP statement that generated the rotate controller or the name of a previous scramble rotate statement.

NOTE: You can skip out of a Rotate block but not into one.

FIX and GROUP Commands

You may have the need to keep some questions in the same order, or in the same position, even though they are within a Rotate block. The FIX and GROUP compiler commands let you control a portion of the rotate sequence. Both commands keep a series of questions together, independent of any rotation. All questions following the commands may be controlled by it (depending on whether a number in block is specified), up to the next relevant compiler command (END_ROTATE, FIX, or GROUP) at the same rotate level. Or, the command may specify the number of questions to keep together as a set. (Any subsequent questions get grouped according to the number specified on the ROTATE command.)

The FIX command fixes the position of a series of questions in a Rotate block. If, for example, the first question in a Fix block is the third question in the rotate, then that question, and the others in its block, will always be asked in the third position.

The syntax for a FIX command is:

```
{!FIX,number in block}
```

If the number in block is omitted, all questions up to the next relevant compiler command at the same rotate level are fixed. If the number in block is specified, that number cannot include other rotate commands.

The GROUP command keeps a series of questions together, without fixing them in the same position each time (as the FIX command does). The questions will be asked as a group, in the same order each time, but not in the same position in the rotate block.

The syntax for a GROUP command is:

```
{!GROUP,number in block}
```

If the number in block is omitted, all questions up to the next relevant compiler command at the same rotate level are grouped together. If the number in block is specified, that number cannot include other rotate commands.

If you have a nested rotate block (i.e., a rotate block within a rotate block), you can only specify a number in block for FIX or GROUP at the lowest (innermost) level.

CfMC recommends that you put the GROUP command around all sets of questions and avoid use of specific numbers per group on ROTATE statements. This gives you more flexibility, allowing quick last minute changes. Here is a sample GROUP command, within a Rotate block:

```
EX:
{!ROTATE,S}
{FIRSTQUE:
.
{SECONDQU:
.
{THIRDQUE:
.
{!GROUP,5}
{STREET:
WHAT IS YOUR STREET ADDRESS?
!VAR,,25,1 }
```

```
{APARTMNT:
IS THERE AN APARTMENT NUMBER?
!VAR,,6,1 }
{CITY:
WHAT CITY IS THAT IN?
!VAR,,15,1 }
{STATE:
WHAT STATE?
!VAR,,2,2 }
{ZIP:
WHAT IS YOUR ZIP CODE?
!VAR,N,5,5 }
{NINTHQUE:
.
{TENTHQUE:
.
{ELEVENTH:
{!END_ROTATE}
```

In this block, note that the specified rotate type is S—this is a scramble rotate. There are a total of eleven questions in the block. The questions will be asked in scrambled order except for the five questions following the GROUP command—these questions will be asked in this order every time. However, the block may appear on the screen as the first question, the fifth question, etc.

With this example, the questions' position within the Rotate block is not important. However, their positioning could be controlled with a FIX command. If the above GROUP command were

instead a FIX command, the series of five questions starting at STREET would always be asked as the fourth through eighth questions regardless of the order of the other questions.

You can have Rotate blocks within other Rotate blocks, up to five levels deep. Each of the ROTATEs requires its own END_ROTATE. The first END_ROTATE will close off the closest ROTATE. There is virtually no limit to the number of questions in a Rotate block.

3.2.2 Compose Control Commands

{!AUTO_PUNCHES}

Causes CAT questions to be created as CAT* questions. This affects the CAT question response list in the following manner:

- Allows only or 1-9/0/X/Y or 01-99 or 001-240 as response codes (depending on the width of the field).
- The response code determines the punch category—a code of 25 is the 25th punch from the start of the question, which at 12 punches per column is the first punch of the third column.
- Data column, offset and recode to punch are disallowed since these are now controlled by the response code. You may still mark items as being exclusive mentions, or designate SKIPTOs.

Since the default is to use the next available punch, this is very useful if you are matching an existing code list. This can be turned off during a session with **-**AUTO_PUNCHES. AUTO_PUNCHES will act as a compiler command from a spec file and put out CAT* and default to the width needed by the question, and error check the response codequits. In the spec file, AUTO_PUNCHES is simply an expedient way of marking a series of questions as CAT* questions. The QSP file will have the CAT question in the AUTO_PUNCHES block written out as CAT*, but the compiler command itself will be commented out.

In the Script Composer, AUTO_PUNCHES acts as a compose mode command. It can be turned on globally to affect all CAT questions composed during the session or for individual questions.

Once you have compiled, all CAT questions composed in AUTO_PUNCHES mode will be indicated by CAT*.

{!AUTO_RESPONSE_CODE=#}

This option will generate response codes for items in response lists (they will be specifically shown in the QSP file). The codes will start at 1. This can make it easier to write up the questionnaire, but use caution when making changes to a questionnaire when you have already collected data.

Option	Description
1	will return codes 1-9, 0, A-Z to a maximum of 36 codes
2	will return 01-99
3	will return 001-500

{!BLANK_LINES=#}

This option puts # blank lines after every question's text. # must be an integer from 0-7. For CAT and FLD questions, blank lines are inserted between the question text and recode table. For NUM and VAR questions, blank lines are inserted between the question text and the prompt.For TEX questions, blank lines are inserted between the question text and the response box.

{BLANK_LINES=0}

This resets this command to the default.

Note that if you specify this command in either your spec file or in the Script Composer, the blank lines will not appear in the QSP file, but they will show on the interviewer's screen. The BLANK_LINES command is passed along to the QSP file and only affects the QFF, HRD, and DB files.

3.2.3 Compile Control Commands

{!COMMENT}

Leaves information after the !COMMENT but before the } as a program comment. See *3.3.2 FILE ACCESS COMMANDS, Using Comments in Specification Files* for details on this command.

{!CHECK_COLUMN_OVERLAP}

Turns on PREPARE's checking of data column overlap. This is the default. -CHECK_COLUMN_OVERLAP would turn the checking off.

{!DEFAULT_NUMERIC_MINIMUM=xx}

This changes the default minimum value for numeric questions that have not had a minimum value specified. The options are:

Option	Description
0	Sets the default minimum to "0" (default).
MINIMUM_NEGATIVE	Sets the default to the minimum value allowed by the width of the question; i.e. 0 if the length is 1, –9 if the width is 2, -99 if 3.
1	Sets the default minimum to "1".

{!DO_MENTOR}

Does both a {!DO_VARIABLES} and a {!MAKE_SPEC_FILES}.

{!DO_VARIABLES}

Turns on the making of variables in the DB file. This is the default. Use -DO_VARIABLES to turn it off. If the header option -DB_FILE is specified, you will get no variables regardless of this setting.

{!ERROR_STOP #}

Tells PREPARE to stop compiling after the number of errors specified is encountered. The 'Finished compile' message prints with the number of errors found up to that point.

{!HARD_CODE}

Tells PREPARE to hardcode the data locations in the QSP file. If you use this option before starting to interview, and then use the QSP file to make any changes, it will prevent you from overwriting previously used data locations. You may turn this on or off for different sections of your questionnaire. Turn this off with -HARD_CODE.

{!HARD_COPY option**}**

Causes a hardcopy file to be made. The hardcopy file will be called <studyname> with an extension of HRD (see >PRINT_FILE <name>, USER to override this default name). You can turn this on or off anytime in the questionnaire. Use -HARD_COPY to turn it off.

Additionally, the HARD_COPY command has optional parameters to control the output HRD file listing. Enter the options after each other, separated by commas. To change the option's default, put a dash (-) before it (i.e., HARDCOPY -SHOW_CAT_AS). You cannot continue a HARD_COPY command to another line. If needed, have multiple HARD_COPY commands on separate lines.

Option	**Description**
ALL	Prints all questions, including GOTOs and RESETs. Default is -ALL
COMMENT	Writes your {!COMMENT} parts of the questionnaire to the hardcopy. -COMMENT turns this off (default).
DATE	Prints current date. Default is -DATE.
LANGUAGE=xx	Prints a hardcopy script in the language specified when you have a multi-language questionnaire. (See *2.5.5 MULTIPLE LANGUAGES* for more information on specifying languages.)
LINES=#	Skips # number of lines after the next question.
LINES_BETWEEN_QUESTIONS=#	Places a graphic line between questions (default).
ONE_PER_PAGE	Puts each question on its own page. -ONE_PER_PAGE turns this off (default).

PAGE	Allows new page break here.
PAGE_LENGTH=#	Allows you to specify the default page length. This is ignored in favor of fitting the whole question on a single page.
PAGE_TITLE=x	Prints title on TOP or BOTTOM (x). Default is TOP.
PAGE_TITLE="text"	“Text” replaces the study code and study comment on the page title. If used in combination with PAGE_TITLE=TOP or BOTTOM, you must specify this as PAGE_TITLE=“text”, PAGE_TITLE=TOP. Your default PAGE_TITLE prints as ‘Date - study code -study comment - page #’.
PRINTER= 0	Prints with no printer type set—strips enhancement escape sequences. This is the default. 1 Prints with HP Thinkjet type set; 2 Prints with HP Laserjet type set
RESPONSE_INFORMATION	Shows info about length of response, how many, etc. (not the default)
RESPONSE_ITEM_LEADER_DOTS	Puts leader dots between response code and text (default)
RESPONSE_ON_RIGHT	Puts response to the right of the code (default)
SHOW_CAT_NS	Shows CAT,N and FLD,N response lists. Default is not to show them.
SHOW_DATA_LOCATIONS	Show the data location (default)
SHOW_LABELS	Shows question label; does not print for GEN questions. Default is to show if there is a user-supplied label.
SHOW_LOGIC	LOGIC Shows logic (IF) of question. Default is to show.
SHOW_QQ_NUMS	Shows question number; does not print for GEN questions. Default is to show, even for program-supplied questio numbers.
SHOW_SAMEAS	Shows SAMEAS response lists. Default is not to show them.

Here is example default output:

```
September 17, 2004 - Exam1 - Example Questionnaire - Page 1
____________________________________________________________
Question: CNTRY

       Countries
(50.4)  01 ............ United Kingdom
        02 ............ France
        03 ............ Germany
        04 ............ Spain
```

{!STRIP_HTML}

This statement strips HTML code out of the text of the questionnaire, so that users can generate support files without the accompanying HTML code. The QFF file, QSP file, CHK file, SUM file, HARDCOPY file, and DB file are all affected by this. This allows you to generate reports for CfMC Utilities (SCAN, LIST, FREQ), word processors, or ASCII editors without the html code being exported. Or, you can convert the questionnaire to a standard CATI questionnaire.

NOTE: Since the .qff file is affected, be sure to save off the running .qff file, or use the command "QFFNAME=NOHTML" to save the .qff file with the name nohtml.qff, so you don't mistakenly try interviewing with it.

Also see "~specrules strip_html" which only affects the .def file and .db file variable text and not the other files. This would be useful for stripping the html for tables only.

Here is some additional information on how different statements are handled:

- The question number or label does not print for GEN questions.
- The col.width does not print when width=0 (e.g., DISPLAY question).
- GOTO questions with text will print in the HRD file.
- CAT questions in Autopunches mode (CAT*) print the recode table in double caret (^) mode.

IF STATEMENTS:

The translation of the IF statement works for CAT and FLD questions only. The translation picks up the response text of the item(s) referenced in the expression in an attempt to put out an English-like logic statement. If the spec looks like this,

```
EX:
!IF DRINKS(04,09,13,16,22,25,28,34,39,42)
```

then the hardcopy file would look like this:

```
            IF: (DRINKS IS DIET COKE or DIET DR. PEPPER or DIET
            YELLOW MELLOW or DIET MOUNTAIN DEW or DIET MUG
ROOT                   BEER or DIET MUG CREAM or DIET PEPSI or DIET
7-UP

            or DIET SLICE or DIET SPRITE)
```

Note that only the first 20 bytes of recode text are picked up. Long recode text will be truncated accordingly. You can override PREPARE's IF translation by adding your own comment

```
!IF |comment| expression
```

Use this in any IF/EXP you wish to document differently in the HRD file.

{!HIDE_ALL}

Hides all subsequent questions. This is useful when changing the questionnaire for resumed interviews. Use -HIDE_ALL to turn off. Questions being hidden are still checked for syntax, logic, etc.

{!INCLUDE filename **}**

Specifies the name of a file to be included in the compile. The specifications in this file are not expanded into the Script Composer question list. See also &filename in *3.3.2 FILE ACCESS COMMANDS, Getting Specifications From An ASCII File*.

{!MAKE_SPEC_FILES}

Turns on the making of specification files (Mentor, SPSS, etc.). This is the default. Use -MAKE_SPEC_FILES to turn off. This does not affect the making of variables for the utilities (see

DO_VARIABLES). This command has no effect on the QSP, CLN, SUM, or CHK files. You must be making the spec files for this command to have any effect.

{!MENTOR_CLN_FILE}

{!MENTOR_DEF_FILE}

{!MENTOR_TAB_FILE}

These commands control the information that gets passed to the CLN,DEF and TAB files. (See your *MENTOR Manual, 4.11 USING PREPARE TO GENERATE MENTOR SPECIFICATION FILES* for more information.)

{!NUMBERING #,#**}**

Controls incrementing for PREPARE's automatically assigned question numbers. The first # is the question number which will be used for the next question. The second # is a number that will be the increment between question numbers. Either the next or incrementing number is required. A next question number is not required—if you don't use it, you must include a comma as a placeholder, and PREPARE will use the last known number and increase it by the increment. This command controls all subsequent questions until another NUMBERING command appears. The default increment is .02. The default starting question number is .03. The increment will affect questions (CAT, GEN, GOTO, etc.) but not compiler commands, which *do* use up question numbers. These will increment by .01.

{!REMOVE}

{!END_REMOVE}

Prevents compilation of a block of questions, comments, etc. REMOVE requires an END_REMOVE command to indicate where the block ends. You can also nest these commands (there is no known limit). This can be used to remove questions from an existing questionnaire. Things being removed are not checked at all for syntax, logic, etc.

NOTE: Meta commands within the block will not be removed, and they will *not* be executed. All other questions and commands will be ignored within the !remove/!endremove block.

{!RESPONSE_ON_RIGHT=x**}**

This command should be used if you have specifications for CAT or FLD response lists from an outside source where the response code appears to the right of the response text (e.g., Apples 01). If you have some repeating character (e.g., ".") between the response text and the response code, you may specify that character in the command.

```
EX:
{!RESPONSE_ON_RIGHT=.}
{FRUIT:
What's your favorite fruit?
!CAT
Apples .... 01
Bananas ... 02
Cherries .. 03
}
```

The response list will appear in the interview as a normal CfMC response list:

```
What's your favorite fruit?
01 Apples
02 Bananas
03 Cherries
```

Any extra blanks beyond the mandatory single blank between the response text and the response code will be suppressed automatically. The "=" option need not be specified. Therefore, if only {!RESPONSE_ON_RIGHT} is specified, only extra blanks will be suppressed (along with reversing the order of the code and the text).

{!SET_QUOTA name or number**=#,OVERRIDE}**

Sets the initial quota value in new quota files. This command will not affect existing quota values unless the ",OVERRIDE" parameter is used.

{!TARGET quotaname**=**#**}**

Sets the value for the Target quota quotaname.T; # is 1-99999. It is used in comparison to quotaname to determine whether quota has been met.

Usually one TARGET command for every quota (see *3.1.5 SYSTEM INFORMATION STATEMENTS, Using Triplequotas Mode*) is used.

TRIPLE_QUOTAS must be set in the study header. This command will not affect existing quota values.

{!TARGET <name=value>, OVERRIDE**}**

This directive to set the Target value will cause the specified value to be saved even though the quota already exists and has a value. Without “override,” an existing quota value is not changed.

3.2.4 Data Control Commands

{!BLOCK #**}**

{!END_BLOCK}

This marks a block of questions — the total number of columns allocated to these questions is specified (#). This is useful when you have a set of questions that may be expanded later, so you reserve the extra space with this command.

{!COLUMN_KICK #**}**

This causes # blank columns before the next question's response. It is useful for reserving columns for future added codes or questions.

{!COLUMN #**}**

This causes the next question's answer to be assigned to column # (a numeric entry). Use with caution to avoid overlapping responses. If you continue to add questions without data locations after this command, they will skip over any questions previously recorded there.

{!HIGH_POINT x**}**

This saves the current location as *x* (a letter a-z, so you may save up to 26 locations) if it is higher than a previous one saved as *x*.

{!RESET_COLUMN #**}**

This removes references to existing question locations after the column specified (#). Use this command to reuse a data area over and over without worrying about specifying a location on each question.

{!RESTORE_COLUMN x**}**

Resets the current location to the column saved as *x*. Subsequent questions are assigned columns beginning with the restored location.

{!SAVE_COLUMN x**}**

Saves the current location as *x*. You can then return to the location for subsequent questions later in the questionnaire by using {!RESTORE_COLUMN x}.

Here is an example using HIGH_POINT, RESTORE_COLUMN, and SAVE_COLUMN:

```
EX:
{APPLE: 20
Do you like apples?
Y or N
!VAR,,1,1}
{!SAVE_COLUMN A}
{500
!IF APPLE$="Y"
Why do you like apples?
!TEX}
{!SAVE_COLUMN B}
{!RESTORE_COLUMN A}
{YESNO:
Do you like oranges?
Y or N
!VAR,,1,1}
{!HIGH_POINT A}
{!RESTORE_COLUMN B}
{
!IF YESNO$="Y"
Why do you like oranges?
!TEX}
{!HIGH_POINT B}
{!RESTORE_COLUMN A}
```

In the above example, SAVE_COLUMN A stores the current location (20) as A. The next question uses column 500. The SAVE_COLUMN B stores the current location (500) as B. The "Do you like oranges?" response will be stored after the "Do you like apples?" location using the RESTORE_COLUMN A command. The HIGH_POINT A saves the current location (21) as A. The "Why do you like oranges?" response is stored after the "Why do you like apples?" location using the RESTORE_COLUMN B command. By following this scenario, you can organize your data file for your own purposes, although using the work area (see Header option WORK_START) may be easier.

{!START_NEW_CARD}

Forces the data for the following question to the first available position on the next record. This is useful in column-free mode where you want to start data for a certain section on a new record.

REFORMAT DATA CONTROLS AND USE

You can control the output of the spread data by using compiler commands in the PREPARE program. Remember that you can override some of the CAT reformat commands from options in the REFORMAT program. Note that these compiler commands do not affect the flow of the interview in any way and can be inserted or deleted after interviewing has been completed to change the format of the output file.

Following is a description of each of the commands, and what they control.

{!EXPORT_LEVEL=#}

This says that variables after this statement get marked with a level that can be used when using reformat to determine which variables to include. If the export level is <= the level specified in the reformat run for a particular question, the data from that question will be included.

{!RFT_ON}

Includes the following questions in the RFT file; this is the default. To exclude questions (like unwanted internal calculations) from the RFT file use the command {!-RFT_ON}.

{!RFT_CAT_01}

Controls how CAT questions are spread. One column is used for each possible response code (in the order that codes appear in the recode table), with a 1 if the code is present, and with 0 if the code was not. This is the default but it can be overridden with option 5 in the REFORMAT program. Option 4 in the REFORMAT program will override this for single response CATs.

{!RFT_CAT_PUNCH}

Puts out single or multiple-response CAT question data as punches 1-9, 0, X, and Y in the RFT file. For multiple-response CAT questions the number of columns is determined by the original width of the question times the maximum number of responses allowed (will be more for the LOTUS format since this adds a comma between data items). For example, if your original question uses three columns and maximum responses is four, then this command will put out four three-column sets for a total of 12 columns in the spread data file (and 16 if you use LOTUS format). Responses are spread as punches (one punch per column set) in the order that they appear on the recode table. The position of a punch in its column set is determined by its column position relative to the start or base column.

{!RFT_CAT_RESPONSE}

Saves space only for the maximum number of responses allowed, and puts the response codes directly in the data, in the order they appear on the recode table. You can override this for single response CAT questions with option 4 in the REFORMAT program which will put out the punch instead.

{!RFT_CAT_SPREAD}

Spreads multiple-response CAT data as punches, sequentially, in the recode table response order as one punch per column in the spread data file. This is like the default of '1s' and '0s' except that each column will either have the original punch or be blank.

{!RFT_SAVE_LOOPS}

Saves loop data in the order that it was entered, for the number of columns needed to hold each response set. It only saves enough data columns in the RFT file for the MAXIMUM number of times through the loop (which can be fewer than the number of possible LOOP controlling responses).

{!RFT_UNWIND_LOOPS}

Generates columns in the RFT file for every possible iteration of the loop and puts the data in its proper columns in the loop according to the response code it is related to. For example, the third response (from the LOOP controller) would have its data go in the third loop data space in the RFT file. This will use enough columns in the RFT file for the number of responses in the LOOP controller (which may be more than the maximum number of responses in that question). Note that there will be gaps of blank space in the data for each response that was not mentioned. This is the default.

Once the data has been spread, you can then read the RFT file into programs that only understand ASCII input or send the data to someone who will be using such a program. If only the closed end data is required, then suppress the creation of any header or text records.

The TEX question output can be used to either generate a response list and recode the data using the RECODE utility or coding mode, or input it into another program for additional processing (spell-checking, for instance). If you are just generating a list of responses to the TEX questions, then the LIST utility might be more beneficial.

If you use ~REFORMAT SPSS, SAS, MENTOR, etc., you will *always* get certain parameters, e.g. CAT_RESPONSE questionnaire parameters such as RFT_CAT_PUNCH will be ignored. Many of the other keywords will be ignored in this case. Same if you say CAT_RESPONSE_01 in REFORMAT.

3.2.5 Help Compiler Commands

Help compiler commands put a HELP message at the bottom of the interviewer's screen for the question type(s) you specify. There are default messages. For instance, for NUM questions, it puts the message "Enter a number from 1 to 10 or DK" for a question with a range 1-10 and exception DK. It also allows you to put a help message of your choosing of up to 80 characters at the bottom of the screen, based on question type.

For example, {!HELP_NUM,Enter a numeric response} would put the text "Enter a numeric response" at the bottom of the screen. Note that you can use back-references to other questions in the !HELP text to show their values as part of the display.

The syntax for !HELP is:

```
!HELP_<question type>,<help message>
```

The question types you may use and their default text are:
(Remember that the _ is optional)

CAT	Enter <up to #> response codes, press **Enter** to continue
DISPLAY	Press any key to continue
ECHO_M	Enter 'item' to select, '-item' deselects. **Enter** to end
ECHO_S	Enter 'item' to select, **Enter** to end
EDIT	Edit text, "H" for help, "M#" to modify line #, "E" to continue (DOS) Edit text, use arrow keys to move, use **DEL/INS**, press **ESC** to end (UNIX)
FLD	Enter <up to #> response codes, press **Enter** to continue
FLDX	Enter <up to #> codes using '+ item' to select, '-item' to omit.
GRID	Use arrow keys to move, ^ to back out, **ESC** to end (DOS) Use **Ctrl-D/U/R/L>** to move, **ESC** to end (UNIX)
HILITE_MULTI	Use 'item' or arrow keys to move, INS/DEL or SPACE to select/omit (DOS) Use 'item' or **Ctrl-D/U** to move, **Ctrl-N/P** or SPACE to select/omit (UNIX)
HILITE_SINGLE	Use 'item' or arrow keys to move, press **Enter** to select (DOS) Use 'item' or **Ctrl-D/U** to move, press **Enter** to select (UNIX)
NUM	Enter a number from <min> to <max> <or 99, DK, YY>
RSET	Press **Enter** to reset to <label> and re-enter response

TEXT	Use arrow keys to move, **DEL** to delete, **ESC** to end Enter your response text, press **Enter** twice to continue (UNIX line mode)
VAR	Enter from <min> to <max> characters
VAR_L	Enter from %d to %d alphabetic characters
VAR_N	Enter from %d to %d numeric characters

```
EX:
!HELP_CAT,Enter one response only please
```

!HELP on a particular question overrides these defaults (see *2.8 ADDING YOUR OWN HELP MESSAGES*). Within a GRID, the HELP compiler commands are turned off except for HELP_GRID. !HELP on a particular question overrides the HELP_GRID, so for each question you move to on a grid, you can get other help information.

Use {!-HELP} to turn off all help lines in the questionnaire.

3.3 PROGRAMMING ENVIRONMENT

With PREPARE, you can do many things to control the programming environment:

- Control access to previously created questions from a DB file
- Control access to specifications in ASCII files
- Access operating system commands
- Access CfMC specific program control commands

Meta commands are used to perform many of these special functions. The meta character (>) is specified, directly followed by the specific command. For meta commands that turn on or off certain functions, you can use a minus sign (-) after the meta sign (>) to turn them off.

There are also special ways to reference files while in the PREPARE program using ampersand (&) file name references. We will very briefly examine each of these types of access below. See the *Utilities* manual for more information on these commands.

3.3.1 Data Base Creation

PREPARE automatically opens a DB file to store the converted variables (for Mentor and the utilities) in. The commands to start a study and store the variables as they are created are:

```
EX:
~PREPARE COMPILE   ''Start PREPARE session
[NEWST]            ''Study header
```

This will automatically create a DB file called NEWST with a DB extension. Unless -DB_FILE is specified in your header tatement, your variables will be converted and stored in the DB file.

In addition to the defined variables, PREPARE creates variables to store the questionnaire header statement <studyname>_HEADR) and case id (<studyname>_CASID).

3.3.2 File Access Commands

>PURGE_SAME

Purges any existing file (except the file named at the List File prompt and the QUO file), that has the same name as any new file created by the program. Without >PURGE_SAME, the program renames old files by changing the file name's first character to the next ASCII character. Use >- PURGE_SAME to return to the default of not overwriting files.

The syntax is:

```
>PURGE_SAME
```

See the *Utilities* manual for additional commands >EDIT_FILE, >END_OF_FILE, >FILE_TO_DB, >SAVE_AS_FILE, and >SAVE_KEYS.

Getting Specifications from an ASCII File

Data entry questions, questionnaire control statements, groups of questions, header information, and large response lists from previous questionnaires can be brought into a current questionnaire.

When a questionnaire is compiled, PREPARE automatically saves all the specifications in an ASCII (QSP) file. You can copy an entire ASCII file, or any portion of one, into a current questionnaire — editing it as needed.

The syntax for reading in a specification file (at the PREPARE--> prompt) is:

```
&filename
```

Filename is any valid file name. With the above syntax, the specifications in the file will be read into the program and displayed in the list file.

You may want the file brought in without displaying the specifications in the list file. The syntax for this is:

```
&-filename
```

&? says that if the file referenced is not found, skip over it and don’t generate an error. Another option is to specify just a portion of a file, by using the slash (/) delimiter:

```
    &filename(parameter 1/parameter 2)
```

This syntax is used to indicate that you want only the part of the file from the first occurrence of the first parameter to the next occurrence of the second parameter. A parameter can be a question number, a string of ASCII text or a line number.

Specification Parameters

To locate the beginning of questions, PREPARE places a commented question number before each question in the QSP file. CfMC’s comment character is the apostrophe. Commented question numbers take the form '###.##. For example:

```
    EX:
    &HOSPITAL.QSP('0.05/'1.10)
```

With this syntax, PREPARE would bring in everything from the beginning of question .05 to the beginning of question 1.10. PREPARE is searching for this comment beginning in column 1 of your spec file.

Because commented question numbers precede the questions, it is important that the second parameter be beyond the last question desired. In the above example, only the comment line before question 1.10 would be included.

You may also reference a file by line number, or specific characters in the file. See the *Utilities M*anual for more information.

Using Comments in Specification Files

There are many ways to put comments or comment out items in a file. You can put comment lines in your specification files for your own documentation. A simple way to do this is by using apostrophes (''). Two apostrophes may be used to comment out everything to the right on that line. For example:

```
    EX:
```

```
'' The next question has to do with credit cards
{
Do you use credit cards? ''question text
!CAT ''question type
1 Yes ''first response code
(SKIPTO services) 2 No ''second
(SKIPTO services) Y DK/NA} ''last line of question
```

In this example, all the text on the right (following the double apostrophes) will not affect the specifications. They are simply explanatory notes. They will not appear on the screen during the interview. Make sure you use two single apostrophes, not one double quote.

These comments will be passed to the QSP file. You can also annotate spec files with comment blocks that PREPARE will pass to the QSP file. The syntax for using comment blocks in spec files is:

```
{!COMMENT
<text lines>}
```

The format is similar to other compiler commands but you can have approximately 100 lines of text and use any special characters inside the block. You can comment out a whole question by placing a {!COMMENT in front of the question's open {, just do not add another closing } or you will get a spec error. As soon as a closing brace is seen, the Comment block is ended. If you need to create a Comment block around a group of questions, use the REMOVE and END_REMOVE compiler commands.

You do not need an ampersand (&) to continue to the next comment line, just type as you would normally. The closing } brace can be at the end of the last line of text or on the next line.

NOTE: COMMENT must come after the study header. If you want comments at the top of your spec file use the apostrophe comment characters but remember these lines will not get passed to either the QSP or HPS file.

Here are some examples of how comments are used in the spec file.

```
{!COMMENT only one line of text}
{!COMMENT           text does not need to start on same line
text line 1
"text line 2"       special characters okay
}                   ends this Comment block
```

NOTE: Meta commands in column 1 are dropped from the Comment block passed to the QSP file and do not execute. Here is an example:

Spec file: **QSP file:**

```
{!COMMENT           {!COMMENT}
>DIR
}

{!COMMENT >DIR}     {!COMMENT >DIR}
```

You may also comment out any question or compiler directive by using a minus sign (-) after the open-brace that begins the item:

```
{-Q3: minus sign causes this question not to
      execute
This is the text of the question
!FLD
1 one
2 two
}     ends this 'commented out' question
```

3.3.3 Operating System Commands

>SYSTEM Commands

Operating system commands may vary from system to system, but you can call any operating system level command or program by first specifying a "SYSTEM."

Operating system commands can be called from within the program. These include TYPE, DIR, LISTF, TREE, MD, RD and program calls. (See your operating system command guide for more information.)

```
EX:
>SYSTEM DIR *.TR
```

NOTE: The Meta commands COPY, DELETE, PURGE, and RENAME will work across operating systems (DOS, UNIX). These are CFMC Meta commands and should not have "SYSTEM" specified before them. These can be used in Survent in Debug or View mode. To use the operating system's version of these commands, enter ">SYSTEM <command>"

3.3.4 Program Control Commands

>BROWSE

Displays a file on the screen. It will automatically pause when the screen is full.

```
EX:
>BROWSE myfile
```

These affect current interaction with the program, screen and file display, and specification production.

>CASE_SENSITIVE (UNIX)

Tells the program whether to read filenames in UNIX the way you type them in the program, or whether to assume no matter what you type, they are lower case names. The default is that it assumes they are lower case names.

>COLORS

Controls the screen colors for any CfMC program, overriding the system's default colors. You can specify three different color specifications on this command: the screen colors, the color for question text, and the color for response text. The question text screen would always be displayed in the colors specified as the second and third color specs.

In DOS, >COLORS also sets the default colors for Survent interviewing or CfMC utility programs. This command can be specified in the INITIAL file (\CFMC\CONTROL or CONTROL.CFMC) or at a CfMC program command line. The >COLORS command overrides the SET COLORS= environment variable.

To use colors in UNIX you must have "setenv COLORS ON" or >COLORS ON in the initial file.

See section *2.5.2 TEXT ENHANCEMENTS AND GRAPHIC CHARACTERS, Color Enhancements* for more on specifying colors in your questionnaire specifications.

The syntax for >COLORS is:

```
>COLORS CfbCfbCfb or >COLORS ON

EX:
>COLOR CWCCWRCRW
```

This example sets the default screen color to white on cyan, question text to white on red, and response text to red on white. Use a "+" before the colors to turn bold.

```
EX:
>COLOR C+WB
```

>DEFINE @name text

This says to define a variable name which is a replacement value for the text following. This is often used to shorten the spec-writing time or set standards interviewer prompts, etc.

```
EX:
>DEFINE @num99 Enter a value from 0 to 99
or DK if don't know
```

This would then be used when defining the text for a statement in the questionnaire:

```
EX:
{
How many apples did you eat this week?
@num99
!num,,,0-99,,dk}
```

>FORCE_HARDCODE

This says to "hard code" the data location on all questions in the backup QSP file. If you place this command in the CfMC mentinit file, all backup QSP files will be hardcoded. This is useful if you decide to make changes; you can recompile the QSP file with the changes so current questions do not move to new locations.

>IF_DEFINE @keyword ... >ELSE ... >ENDIF

Says that if the @keyword is defined then do what comes next. Requires >END_IF to end the IF block, and allows >ELSE to do actions if the statement is not defined. You may have nested >IF_DEFINE blocks. Use this as a conditional statement to control branching in a spec file.

The syntax for >IFDEFINE is:

```
>IFDEF @keyword (required)
… (commands) (optional)
>ELSE (optional)
… (commands) (optional)
>ENDIF (required)
```

See also >IF, >ELSE, >ENDIF and the IF/ELSE/ENDIF commands in DOS and UNIX operating systems.

>IF expression ... >ELSE ... >ENDIF

Says that if the expression is true, do the following commands, otherwise skip them. You may use nested >IF blocks and may use it in conjunction with >IF_DEFINE. You may use this with defined keywords as well to check the value of the keyword (note: you will need to set >FILL_DEFINES_IN_QUOTES to use this option with defined keywords).

```
EX:
>IF "@Prodtype" = "Software"
{
Enter the name of the software products you use
!text}
>ENDIF
```

Use = or <> to compare strings. For numeric comparisons, use =, <>, >, <, >=, or <=. You can use "AND" and "OR" for complex expressions.

>HALT

Controls when the program stops and waits for an **Enter** from the keyboard to continue. These commands will work when PREPARE is expecting possible additional input from the keyboard (i.e., almost all cases except when your PREPARE command included your spec file name without an ampersand preceding it).

The syntax for >HALT is:

```
>HALT keyword
```

The keywords are (none), ANY, ERROR, WARN, and NONE.

>LOCATION_FORMAT

Controls the print format of data locations by the program, i.e., allows the user to specify the format of data locations in the CHK and SUM files, error and warning messages, the hardcopy file(HRD) and QSP file question comment line.

The syntax is:

```
>LOCATION_FORMAT format entry
```

where the entry may be one of:

1 Displays the data location as absolute columns.

1/1 Displays the data location as record/column where each record is 80 columns. This is the default.

A01 Displays the data location as letter digit digit. B23 is record 2, column 23. use in conjunction with the SPL_DATA_LOCATIONS_OK header statement option.

>PRINT_REPEAT

This controls whether the result of the expanded >REPEAT sequence is printed to the list file (see >REPEAT). The default is that it is printed. Use >-PRINT_REPEAT to turn off the printing of statements generated by a >REPEAT.

The syntax for >PRINT is:

```
    >PRINT_REPEAT
```

>QUIT

Immediately returns to the operating system prompt from any program. If you use this command, PREPARE will not create backup files, or purge temporary files used by the programs.

The syntax for >QUIT is:

```
    >QUIT
```

>REPEAT/>END_REPEAT

>REPEAT sets a repeating pattern to produce multiples of specifications, using variable names for changing items within the repeated sequence. This can be used to create PREPARE specifications for repeated questions (e.g., rating scales), blocks of questions, or text. The syntax or a repeat block is:

The syntax for >REPEAT and >ENDREPEAT is:

```
    >REPEAT $Name1=Element1a,...,Element1n;
    $Namen=Elementna,...,Elementnn
.
    Text or command lines to be repeated ...
.
    >END_REPEAT
```

Here is an example:

```
    EX:
    >REPEAT $A=1,...,3;$B=Apples,Oranges,Bananas
    {RATE$A:
    How would you rate $B?
    !CAT
    3 Excellent
    2 Fair
    1 Not Good}
    >END_REPEAT
```

RATE$A would produce RATE1, RATE2, and RATE3.

>-SURVENT_DEBUG_INFO

Removes the default information that displays to "debug mode" interviewers at the end of an interview. See the *Utilities* Manual for more information, and additional Meta commands.

3.4 SOUNDSURVENT AND WINSOUND

3.4.1 SoundSurvent Overview

WHAT IS SOUNDSURVENT?

SoundSurvent is an add-on to Survent that allows you to collect or play back sound during an interview. WINSOUND is the Windows version of SoundSurvent.

WHEN WOULD YOU WANT TO USE IT?

Use SoundSurvent when you want to play back sounds as a part of an interview or record responses to open-ended questions. You can also record interviews to monitor an interviewer's technique.

HARDWARE REQUIREMENTS

•A UNIX machine or PC running Survent software.

•A Windows PC with Sound Survent software installed (the SOUND SERVER).

•Up to three 24-port Rhetorex cards installed in the DOS PC.

•An Ethernet connection to a network that the CfMC machine and WINSOUND machine are both connected.

- A telephone for each interviewer, and a telephone connected to an outside phone line for testing purposes.
- Modular duplex phone connector (Y-connector) to split the phone line for each interviewer phone.
- Two RJ-11 phone cables for each interviewer phone.

3.4.2 SoundSurvent Installation

Installation of SoundSurvent is easy. Just follow these steps:

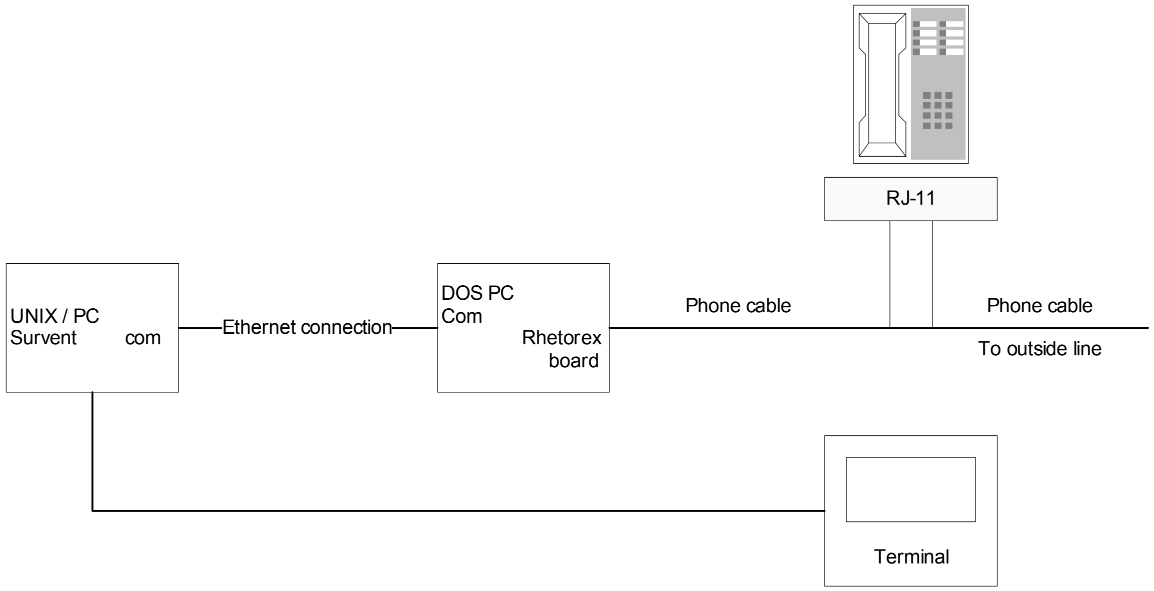
- Connect the hardware (Ethernet connection)
- Modify the PARMFILE and TTYINFO file
- Install the Rhetorex software
- Install the SoundSurvent software

CONNECT THE HARDWARE

1 If necessary, assemble the Survent Server and the Sound Server by connecting the monitors, keyboards, and power lines.

2 Plug the modular duplex phone connector into the back of the interviewer's phone (RJ- 11). Use one phone cable to connect from the Rhetorex board in the Sound Server to the duplex phone connector on the interviewer’s phone. Use the second phone

cable to connect from the duplex phone connector to the outside phone line.



MODIFY THE PARMFILE AND TTYINFO FILE

You must make two modifications: (1) In the PARMFILE, add a line to specify the IP address and port where the CfMC server and SOUND SERVER are connected, and (2) In the TTYINFO file, for each interviewer station running SoundSurvent, you must indicate which sound channel it is connected. Below are more specific instructions for the UNIX platform.

Note: For more information on the TTYINFO file, *please see Chapter 4, section 4.4.4.*

1 Add a line to the PARMFILE (/cfmc/control/parmfile) for each SOUND SERVER. Specify which tty it is connected to, and the high and low number of the ports on the Rhetorex board. For example,

if the DOS PC is at ip address 10.21.222.215 using tcpip port 4000, add this line to your PARMFILE:

```
Remoteserver: 10.21.222.215 4000
```

2 Modify the TTYINFO file to include the sound channel for each interviewing station using SoundSurvent. The TTYINFO has one line per interviewing station; the format of each line is:

```
<term type><tty name><ldev #><ext><time zone>
<snd channel>
```

Below is an entry for an ANSI terminal, tty07, assigned ldev 7, using telephone extension 130, in time zone 8, which is plugged into sound channel 3:

```
ansi   tty07       7     130    8     3
```

In UNIX, you may also use the SOUNDCHANNEL variable in the interviewer login script to set

the channel for SoundSurvent. Just say "setenv SOUNDCHANNEL 3" to set the channel to 3.

You MUST use this if your logins are telnet sessions where the device number is not static.

If possible, configure your phones and software so the phone extension, sound port number, ldev and tty numbers are the same. This will make trouble-shooting easier. For example, this is for interviewing station 10 (in time zone 8):

```
(UNIX)      ansi        10     10     10     8      10
```

INSTALL THE RHETOREX SOFTWARE

1 Turn on the Windows PC and start a DOS box.

2 Load the Rhetorex software from the CD. Enter the following:

```
cd e:\rhetorex
rhetdrv
```

This will load the Rhetorex driver and then display a screen of information.

3 Look for the following headings and values:

```
Status: tested okay
Number of RDSPs Found: 1
Number of Voice Channels: 24
```

If the status is anything other than "tested okay," refer to the Rhetorex documentation.

INSTALL THE SOUNDSURVENT SOFTWARE

1 Load the SoundSurvent software. Enter the following:

```
cd e:\sound
soundsrv
```

2 The program will indicate the number of channels available and that the Rhetorex board setup has completed. Press **Enter** to continue.

3 The top half of the screen will display the status of each channel, and the lower half of the screen will display the commands it receives from the UNIX PC or HP3000. For each channel in use, SoundSurvent displays the a line in the following format:

```
chn   ldev  study       id question time  OP
```

Below is a description of each field:

CHN	sound processor board chanel
LDEV	assigned (logical device) number of interviewer terminal
STUDY	the study name
ID	interviewer ID
QUESTION	the question label of the current question

TIME	starting time of the process

OP	current operation
	A= answer
	C=playback*
	I=Idle
	P=playback
	R=recording
	S=stopped
	T=playback*

* *See the Spec Writing section (3.4.3)* for a description 96 of playback modes.

OPERATION, TESTING SOUNDSURVENT

1 Start Survent Server and Survent Supervisor as you would normally.

2 Pick up the handset of the interviewer phone that is connected to the Sound Server PC.

3 Press any key on the phone to silence the dial tone. At the Survent server, press **Ctrl-Y** and enter:

```
sound:p <port>,<port>,a,b,parrot
```

where <port> is the port number of the phone you are listening to. This should send a message to the SOUND SERVER to play the 'parrot' file. This command should appear on the lower portion of the SOUND SERVER screen and you should hear the sound in the telephone handset.

4 When you are ready to stop playing the parrot file, press **Ctrl-Y** again and enter:

```
sound:s <port>
```

If you cannot hear the test sound file, check the port assignments in the TTYINFO file and verify that the phone is connected to the correct port.

A default file named SNAPPY.TUN is required. This is to avoid problems when there is no sound returned because you requested a sound file that did not exist.

STARTING SOUNDSURVENT

If the SoundSurvent hardware has been correctly installed, your Survent specs will start and stop SoundSurvent for the appropriate questions. All you have to do is start a questionnaire as you normally would.

Start the questionnaire by entering the study name and the interviewer ID. Enter Y or N to indicate if the interviewer ID is correct. Press **ENTER** at the usual Survent questionnaire prompt:

```
(studyname) Enter <return> to interview, or Quit -->
```

Shutting Down the SOUND SERVER (the DOS PC)

1. Make sure Survent and the Survent-server are shut down.
2. At the SOUND SERVER, enter:

   ```
   quit
   ```

3. At the DOS prompt, turn off the DOS PC.

Automatic Recording Shutoff

The SOUND SERVER identifies when there is not enough disk space to hold new recordings, and stops recording at that time. This is to keep it from crashing when it runs out of space. It sends the estimated recording time left back to the Server with each 'okay' message response. When the amount of time left is less than 20 minutes (default), the recording stops and a warning is issued to the supervisor and interviewer. The time when the recording stops can be changed by sending the SOUND SERVER the command "MINTIME=#". For example, in a SURVSUPR session, enter "SERVER:SOUND:MINTIME=60" to set the cutoff time to 60 minutes.

3.4.3 Spec Writing

Use the following guidelines when writing questionnaire specs for SoundSurvent.

SOUND SERVER FUNCTION

At the beginning of the questionnaire, check that the SOUND SERVER is running with the SOUND SERVER function:

```
{
!IF soundserver()
!GOTO Q1 }
```

The Sound Question Type

To write questions that record an answer or play a sound, use the "Sound" question type. Sound questions use the following syntax:

```
!SOUND,subtype,option
```

Sound questions have the following subtypes:

R record. The sound file name format is:

```
        S<Julian date><hhmmss in base 36>.<ldev in base
36>
```

This guarantees unique names. You can find out the meaning of the base 36 value by entering “xs<value>” in the MAKECFG program. The hours/minutes/seconds should closely match the timestamp on the file.

S stop recording

A,n answer after n number of rings. This command first hangs up the soundboard and then puts it in answer mode. A,1 will hang up the sound board, put it in answer mode, and then answer after one ring. A,0 will hang up the sound board and take it out of answer mode.

D,x sends dial strings through a sound channel. This can be used to transfer a call to another interviewer or an automatic voice recording system. x is the label or location to send; i.e., a string of characters to send to the soundcard. Any key on the phone pad can be sent, including a comma (,) to pause.

M rename the sound files associated with a specific question by changing the first letter of the filename to the letter specified in the command. Sound subtype M uses the following syntax:

```
!SOUND,M,<label/location of old name>,<letter>
```

N rename a sound file to a specific new name. The name can include a directory reference or disk drive designation to save it in a separate directory or on a separate disk. The syntax is:

```
!SOUND,N,<label/location of old name>,
<label/location of new name>
```

NOTE: When providing filenames in DOS using backslash characters, remember that "\" is a control character in CfMC software; to send a filename use "\^5c" in place of "\" like the following:

```
EX:
{Filename:


F:\^5cSound\^5cMystudy\^5cQst1.SND
!SPC,9 }
{!SOUND,N,Qst1snd,Filename}
```

If you are renaming a file recorded by Winsound, you need to prepend the directory info to the filename as follows for a study called "getsound":

```
{!q1r: !sound,r}

{q1:
Recording sound for q1, stop sound when ready
!disp}

{!sound,s}

{q1fname: .40
```

```
 c:\^5csoundata\^5cgetsound\^5cq1r\^5c\:q1r:
!spc,9}

{q1toname: .40
c:\^5csoundata\^5cgetsound\^5cq1r\^5cmyfilename.xxx
!spc,9}

{!sound,n,q1fname,q1toname} ''This renames the file
```

NOTE: Any filename that Winsound sees gets "c:\soundata" prepended to it unless there is a : in the filename already (eg. c:, a drive reference).

For a recorded file, this results in a file named c:\soundata\\\ being created.

For all other filenames \\ is never prepended but if the filename does not have a : then c:\soundata\ gets prepended.

For the !sound,n command the "new" filename is treated as follows:

If it has a : then the new filename is used as is. Otherwise, if the newfile name has a \ then "c:\soundata" is prepended to it. Otherwise the newfile name is just a filename and the complete path, including drive, is taken from the original filename and prepended to the newfile name.

```
      Examples:

c:\test\mine.fle c:\new\mine.fle uses the names as provided

c:\test\jim newone             uses c:\test\jim and makes
                               c:\test\newone

c:\test\jim \new\mine          uses c:\test\jim to make
                               c:\soundata\new\mine

\orig new                      uses c:\soundata\orig to make
                               c:\soundata\new
```

X Reestablishes the connection to the sound server. This would be used with the "!IF NOT(SOUNDSERVER())" command at the top of a questionnaire to re-establish contact.

There are three different playback subtypes. They determine how you can use the telephone keys to affect playback (the telephone key functions are defined below):

P play mode, no key control during playback

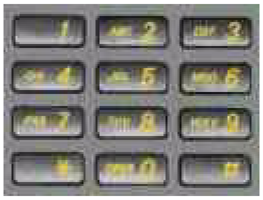
C control mode, can use only keys 7,8,0, and * during playback

T transcription mode, can use all keys during playback

All three play subtypes require a filename. For example:

```
!SOUND,P,sndfile
```

When you are listening to a previously recorded answer (or pre-recorded sound), you can use the keys on the telephone keypad to control playback. The key functions are paired to match the keypad:



1	pause playback
2	backup five seconds
3	go to beginning
4	resume playback
5	skip forward five seconds
6	go to end
7	increase volume
8	speed up playback
9	not used

0	slow down playback
#	not used

Question Labels

You must provide a question label for Sound,R questions. For example:

```
{ Mylabel:
!SOUND,R }
```

Text Alternatives

It is a good idea to include text questions in your questionnaire as an alternative in case there are any problems with the SOUND SERVER. For example:

```
{!If Not(Soundserver())
!GOTO Q2NoSnd }
{ QRec:
!SOUND,R }
{ Q2:
2. Why do you prefer :Q1: ?
\I (Now Recording)\E
\I(Press <Return> to Stop Recording) \E
!DISP}
{!SOUND,S }
!GOTO,Q3}
{ Q2NoSnd:
2. Why do you prefer :Q1: ?
\I (Enter Response and Press <ESC> to Continue) \E
!TEX }
```

Sound File Names

SoundSurvent creates sound files in the SOUNDATA directory with the format SOUNDATA\studyname\questionlabel\filename. The file name starts with an S and is created from the time, Julian date and interviewer ldev. For example:

```
\soundata\widgt\q3sound\s1830947.071
```

This says the file was recorded from the study widgt for question q3sound, and was recorded on julian-day 183 (July 1) at 9:47am by the interviewer at ldev 71.

In addition, there is a parmfile keyword that controls the sound file name. "SOUNDFILE_FORMAT:" controls whether you get a fully qualified name or other formats. This can be specified in the CfMC parmfile or in the questionnaire header. The possible values are:

- EIGHT_DOT_THREE: This will generate a name that the gerry dialer can accept which has Sjjj (julian date), then hhmmss compressed to four characters using base 36, plus a three-character base 36 compressed ldev extension.
- JJJHHMM: This will make a soundfile name Sjjjhhmm. where jjj is the Julian day and hhmm is hour and minute. This is the old default, but there could be a roblem: if you have two files that start in the same minute on the same question, it will overwrite the first with the second, and you can't have an LDEV with a number higher than 999.
- YYMMDDHHMMSS: This is the default using Winsound and will make a name "S._" which has the year-month-day-hour-minute-second laid out clearly. It will also include the interviewer ID (four 4 characters) and ldev (five characters) as part of the extension.

If there is no parmfile keyword, the sound file format will be "EIGHT_DOT_THREE" if you are using the Gerry dialer for sound recording and

"YYMMDDHHMMSS" if you are using Sound Survent. The middle format is to provide back compatibility with the way it worked before 15 June 2005.

Loops

Because of the way sound file names are generated, you cannot use loops with SOUND questions.

Sound File Formats

Rhetorex software includes utilities (CONVERT.EXE) to convert different types of sound files (for example, to change WAV files into SND files). Look under the directory \rhetorex\convert\samples for the utilities and sample files.

CfMC uses Rhetorex's sound recording format because the file sizes are smaller than the alternatives. They are 240 KB per minute, meaning 1GB will handle about 70 hours of recording.

4 CONDUCTING THE INTERVIEW

4.1 USING SURVENT

Survent is CfMC's interviewing program. It takes the QFF file compiled and created by PREPARE, and uses it to present screens to an interviewer, control the flow of the questions, edit the responses and generate associated data. The data file will always have the extension TR.

```
PREPARE --> QFF file --> SURVENT interview -->
Respondent data (TR file)
```

During the interview, the interviewer can enter responses based on question parameters, or they can enter special keywords that allow them to return to previous questions, suspend an interview until later or display prior responses.

Once the data is collected, it can be reviewed in SURVVIEW or LIST on a case-by-case basis. The data can then be modified in SURVVIEW or by using the RECODE or CLEANIT utilities, or MENTOR. Summary reports can be generated using SCAN utilities (or MENTOR).

COPYFILE, REFORMAT, MAKEVARS or MENTOR can be used to create new data files with different file formats, to be used by other software for data manipulation or reporting.

The shared files version of Survent works the same as the standalone version in most respects. It is noted below wherever there are differences.

To start standalone interviewing, enter Survent at the operating system prompt. Standalone interviewing is most often used for testing and for disk-by-mail.

4.1.1 Standalone Survent Startup Under DOS: The Configuration File

The first time you run standalone Survent from a particular directory or drive, a default configuration file is built. This holds information about the interviewing session. The configuration file will have the name SURVENT.CFG. You cannot accept the default configuration, but will have to modify items A and B. Here is the default configuration screen that will appear:

```
1: Interviewer I.D.: intv
2: Data file disk drive: <Current drive>
3: Resume file disk drive: <Current drive>
4: Site code: xx
5: Station #: 00001
6: Show Question #s?: yes
7: Continuous interviewing?: no
8: QFF filename: <none>
A: Study Code: <none>
B: Data File Name: <none>
C: Auto Increment of Case I.D.s?: yes
E: Data file comment: <blank>
F: May Practice interviewing?: no
G: Confirm configuration at start of session?: yes
Item(s) to change or Enter
-->
```

If you want to modify items, enter the corresponding numbers or letters (optionally separated by commas). The program will then prompt you individually for each change. For example, if you enter 1,2,5,A,E, the program will prompt you for a new Interviewer ID, then a Data disk, etc. You will probably want to change at least 1 (Interviewer I.D.), A (Study name), and B (Data file name).

Item D (Next Case I.D.), which is not shown in the initial screen, will show up if C) Automatic Increment of Case ID is set to Yes. If the configuration file has been modified previously and there are no items you need to change, press **Enter** and the interviewing prompt will be displayed. If you already have a SURVENT.CFG file in your local directory, it will be used when you run Survent again.

Here are the items and a brief description of each:

1: Interviewer I.D.: This is used to identify the interviewer in the data; it may be up to our alphanumeric characters long:

```
Interviewer ID (four characters or less)
-->
```

NOTE: If you are testing a phone study in standalone mode, then you can specify special interviewer type and/or practice mode after the interviewer ID.

```
EX: --> Mary,S=13,Practice
```

This would designate interviewer Mary as both a special type 1 and 3 and put her in practice mode.

2: Data file disk drive: This is the drive where your data files will be stored. It will be A or B if working on a floppy drive system, C or D if on a hard drive or possibly E - Z if working on a networked PC system:

```
Data disk (A to Z)
-->
```

3: Resume file disk drive: This is the disk drive where the program will place the files containing any suspended interviews to be resumed. The same rules apply here as in 2 above:

```
Resume/suspend disk (A to Z)
-->
```

4: Site Code: This is any two-character code you want to use to identify the interviewing site. It must be two alphanumeric characters:

```
Site code (two characters)
-->
```

5: Station #: This is usually used to identify which personal computer or terminal was used for this interviewing session. It may be up to four digits:

```
Station number (1-9999)
-->
```

6: Show Question #s?: This controls whether the question number and/or label will be displayed in the bottom right corner of the screen for each question. This is useful to identify the question for the interviewer if you expect them to use the keywords to

show or return to that question, or just to identify which question they are on. Enter Y (YES) or N (NO). NOTE: Interviewers running in Practice or Debug mode will always see this information:

```
Show question numbers? (Y/N)
-->
```

7: Continuous Interviewing?: If this is set to YES, each interview will start automatically without the standard CfMC prompt between interviews. This is used if you wish to write your own interview startup screen. Enter Y (YES) or N (NO):

```
Continuous interviewing? (Y/N)
-->
```

NOTE: Since there is no prompt to Quit from an interview using Continuous Interviewing mode, this must be controlled by using the ABORTOS keyword that will delete the current case and return you to the DOS or UNIX prompt.

QFF file name: This is the name of the questionnaire file created by PREPARE and used by Survent to control the interview. Specify this only if you have compiled different versions of the same questionnaire, e.g., Spanish and English versions. If you are modifying an existing configuration file this item will be filled in by default from the study code specified for item A. You should not enter the suffix, QFF. The program will supply it for you.

```
Questionnaire file name
-->
```

You *must* enter a good questionnaire file name to continue. You can enter just a name if the file is in the current drive and directory, or you can include a disk drive and directory specification if the file is located elsewhere. If the QFF file name is different than the study code you must specify item A first. This field is automatically blanked by the program once you enter the study code. If the CFMCQFL variable is set, Survent will look there for the filename you specify.

A: Study Code: This is the name of the study (specified on the header statement of the specifications file). By default it is also assumed to be the name of the questionnaire file (QFF) created by PREPARE and used by Survent to control the interview. This can be a different name from the actual questionnaire file name as noted under item 8.

This item controls the QUO, FON, FNX, TR (unless item 2 is chosen), and (unless item 8 is chosen) the QFF version to use.

```
Study code (maximum eight characters)
-->
```

You must enter a good study code name to continue and it must be available in the current drive and directory. Study codes may be 3 to 8 characters long.

If you modify item A (Study Code), you will automatically be prompted for item B, and also for item D if that study has a QUO file.

B: Data File Name: This can be any file name. Do not enter the suffix, TR. This will default to the Study Code (see A above) if no name is specified. Remember that the drive specification was set in item 2 above, and should not be specified here.

```
Data file name or Enter for <Study Name>.tr
-->
```

C: Automatic Incrementing of Case IDs?: This allows the case IDs to increment by one for each interview, starting at whatever value is specified. The other option is to have the interviewer (or interview) supply the case ID for each interview. Enter Y (YES) or N (NO):

```
Incrementing Case IDs? (Y/N)
-->
```

If this is set to YES, you should fill in item D below. You may *not* set this to YES if you used option -QUOTA_FILE on your header statement. (IDs are stored in the QUO file.)

If it is set to NO, unless you have used an SPC,A or SPC,D, at the end of each interview, the following prompt will display, with the number of characters determined by the case ID length on your header statement in PREPARE.

```
Enter Case ID (# chars)
-->
```

Enter a case identifier for the interview.

D: Next Case I.D.: If you are using automatic incrementing of case IDs, once a starting case ID has been issued, the program will keep track of the last case ID used, and display the next available ID between interviews, and on the configuration menu whenever a new session is started. The reasons for changing this ID once a job is started include wanting to change to a higher number than previous, or you may be using numbers coded to the site code or station number. The number of characters you are prompted for will depend on what you specified as the case ID length on your header statement. The default is 4 numeric characters.

```
Initial case ID in the data file is: 0001
Enter starting Case ID (4 chars)
-->
```

You can’t set this until after the data file is created when you first run Survent, but you can edit this file (SURVENT.CFG) and place the case ID on line 11. This allows users who send out diskettes to preset all the case IDs that will be returned on individual disks.

E: Data file comment: This comment would display whenever you open this data file with any of CfMC’s programs. This comment will only affect new data files (i.e., once the data file is created, changing this option will not affect the file). If you do not want a comment, press **Enter**:

```
Comment:
-->
```

F: Do Practice Interviewing?: If you respond Y (YES), all new interviews will be started as practice interviews. Data from practice interviews is not saved, quotas will not be saved, and case IDs will not be assigned. Phone numbers will also not be used up in Practice mode. PRACTICE will print in the lower right corner as a reminder to the interviewer. At the end of each interview ***Training Mode*** ***No case written*** will print on the screen to remind the interviewer that they are in Practice mode.

```
Enter Y (YES) or N (NO):
Do practice interviewing? (Y/N)
-->
```

G: Confirm Configuration at start of session?: This determines whether the interviewer will see this menu and be allowed to change any of these items when a session is started. If you respond N (NO), the configuration cannot be changed (unless you rename or delete the configuration file SURVENT.CFG and start the session over again).

```
Enter Y (YES) or N (NO): Confirm configuration?
(Y/N)
-->
```

The file SURVENT.CFG is an ASCII file that can be looked at or altered in an editor or word processing program. This is another way you can change settings if you had set item G to NO.

After making any modifications, you will again be presented with the menu until you are sure that all the items are they way you want them to be.

The interviewer ID, site code, station number, and study name can additionally be stored in the data file using the SPC,7 question type (see *3.1.5 SYSTEM INFORMATION STATEMENTS, Special* Information Statements).

NOTE: *Shared files* Survent has no configuration file. The parameters are controlled by the SURVSUPR program and the employee information file.

4.1.2 Starting Survent (UNIX)

The Survent program is used by the interviewers to collect data. It can be run standalone with the following syntax:

```
SURVENT (DOS, UNIX)
```

DOS versions of Survent present the user with an adjustable configuration menu (See *4.1.1 MODIFYING THE CONFIGURATION FILE*). UNIX Survent prompts the user for only two of the configuration variables, study code and interviewer ID.

Survent can be run within a SHARED FILES configuration by simply using the word SERVER after the program name.

```
SURVENT SERVER (DOS, UNIX)
```

When Survent is started with the SERVER option it is then sharing study files with other userswith the aid of the SERVER program. If multiple SERVERs are running when starting Survent with the SERVER option, you will be prompted for which SERVER you want to use (DOS only). If you try to run standalone and someone is already running standalone on that study, you will get a message and will not be able to run. If someone is already running that job in shared files mode, you will be started in that mode also.

The shared files configuration also includes interviewers being started up by the Supervisor. In this mode, the interviewer acknowledges that they are ready by running the START.BAT file (DOS, UNIX only). See *Appendix G* for more information on this.

Also see *2.2 USING PREPARE, Command Line Parameters* for information on starting up Survent with initial parameters.

When shared files Survent starts up, the interviewers will be asked to identify themselves. They will enter their ID as it is stored in the Employee Information file (see *EMPLOYEE INFORMATION FILE* below).

Interviewer IDs are limited to four characters unless the "LONG_INTERVIEWER_ID: ##.##" parameter is specified in the CfMC parmfile. If it is specified, you put the LONG ID in the employee file in the location specified on the long_interviewer_id command. If this is used, the interviewer can type the LONG or SHORT ID.

After verifying the ID, they will see a message indicating the questionnaire file they've been sent, and then they will see the standard interviewer prompt (see *4.1.4 STARTING THE INTERVIEW*).

At the interviewer ID prompt, the interviewer can override the special type(s) from the Employee Information file by specifying options after their ID:

```
EX:
Enter interviewer ID -->0104,S=34
```

This will assign special interviewer types 3 and 4 to this interviewer.

NOTE: If the Supervisor also issues an S= on the Start command, that will override this setting.

At that same prompt, the interviewer can override the terminal type from the TTYINFO/PARMFILE.

```
EX:
Enter interviewer ID --> 0233,TERM=ANSI
```

The terminal type can be specified as a number or as the terminal type (see *THE TTYINFO/PARMFILE* following).

You can also enter ",PRACTICE","P",",TRAINING", or ",T" after the interviewer ID to put yourself into practice (or training) mode. The supervisor can also put interviewers into practice/training mode (see *4.4.2 SURVSUPR MAIN MENU, STS command*) and interviewers can be assigned training mode in the Employee Information file (see *EMPLOYEE INFORMATION FILE* below).

EMPLOYEE INFORMATION FILE

There is a file that contains information about the interviewers. It is an ASCII file called employee.xxx (DOS, UNIX; this file must be where the variable CFMCCFG points, which is typically \CFMC\IPCFILES in DOS or /cfmc/ipcfiles/ for UNIX). It can be maintained using your editor or word processor of choice. If you wish you may use a different filename in that directory, but each user must have the CFMCEMPLOYEEFILE variable specified with that filename.

This file holds information about the interviewer's ID, their name, and special capabilities such as running in Debug mode, their special interviewer type, and running in Echocats mode.

Parameters are positional by default, but you can use commas as delimiters. The file will be searched fastest if it is sorted by interviewer ID. When sorting the file, note that numbers should be sorted before letters. You'll get a warning if the file is not sorted by interviewer ID.

The CFMCEMPLOYEEFILE variable can point to any filename and it can reside in any directory, provided that all users accessing it have the CFMCEMPLOYEEFILE variable set to point to that name. This allows you to have multiple CfMC environments, all using the same employee file.

Here is the format:

```
    1-4   Interviewer ID(1-4 alphanumeric characters)
      5   blank
   6-30   Interviewer name (in the format you want it to appear on
           their screen for verification)
     31   blank
  32-40   Special types (1-9 in any order)
     41   blank
  42-45   D=DEBUG mode (overrides interviewer controls, allows Alter,
                    GOTO forwards and backwards, can Abort, Break, or
                    Change interviews)
          B=can Abort or Break or Change interviews
          C=can Change interviews

          A=Allows Alter
          N=Interviewer can only interview, and cannot VIEW prior
                    interviews
          O=Interviewer can only interview; disallows practice, debug
                    and dialer modes
          T=interviewer started in Training mode by default
          E=interviewer started in Echocats mode by default
          L=do interviewer logging and save at end of interview
          Q=do interviewer logging and save after every question
46-240   User-supplied employee information or long interviewer id
```

Example:

```
0        1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901234567
1357 John Smith                13        TED John Smith Spanish speaker
```

Your file could also have records that look like this:

```
1357,John Smith,13,TED,John Smith Spanish speaker
```

***NOTE**:*

- There are four special interviewer IDs used by the system. BKOK has BREAK mode automatically; DBUG has debug mode automatically, SV## is used to mark records returned to the sample file from a dialer, and WEBS is used to mark records from a webSurvent session. The ## is the dialer number that made the call; this way you know how many numbers were called by each dialer.
- Use the !SPC,5 statement to retrieve this information in the data. This layout matches that of the SPC,5 statement.

Break Mode and Debug Mode

There are two modes used for testing by programmers. The interviewer ID BKOK or a B in columns 42-45 of the employee file will let you break out of an interview, or enter ABORT to get out or enter GOTO to go to other questions, but all other keywords will only be allowed if they are turned on in the questionnaire or are allowed by default.

Interviewer ID DBUG or a D in columns 42-45 of the employee file will let you do what Break mode allows plus you may use any interviewer keyword regardless of defaults or questionnaire settings. You may also GOTO both forwards and backwards in the questionnaire. The Survent interviewer keyword ENABLEDEBUG will also invoke this mode for the remainder of the interviewing session.

When in Debug mode, the program prints additional information about the recently completed interview to the screen. It will tell you whether a case was written, what phone status was set, what quotas were set, the total session time so far, the interviewer ID, current time, number of interviews started and stopped, and number of completes so far. It also shows quotas and quota changes at the end of a stand-alone interview.

You can turn this information off with >-Survent_Debug_Info, a metacommand. If you never want to see the debug information, add this command to your “initial” file in the CfMC control directory.

Both B and D also automatically allow you to Change (C) interviews. B, C, D, N and O are mutually exclusive; i.e., you cannot use more than one of these at a time.

4.1.3 Standalone Survent Startup Under UNIX

UNIX has no configuration file, just a prompt for interviewer ID and questionnaire filename; you control the interview options relative to DOS in standalone Survent as follows:

1: Interviewer ID: You are prompted for this after specifying the study name.

5: Station number: Is hard-coded in TTYINFO file or by environment variable or specified on the command line when starting a station.

6: Show question numbers: is controlled by the header keyword or compile option -SHOW_QUESTION_LABELS.

7: Continuous interviewing: Use an !SPC,W statement to save the data and !SPC,H statement to suppress the initial prompt in the next interview.

8: QFF file name: By default, you are prompted for this when you start the program or after specifying “CHI” to change studies. If the SHOW_STUDIES command is specified in the CfMC parmfile, a list of available studies is displayed; in this case, you would move to the name of the questionnaire you want to run using the movement keys and press ESC to choose it.

A, B: The study code and data file names are made using the study code in the questionnaire header. You cannot write to a data file that has a different name.

C: The Header keyword -AUTOMATIC_CASEID_INCREMENT controls automatic increment of case IDs.

E: There is no way to set the data file comment in Survent, you can pre-build the file using COPYFILE or MENTOR (~INPUT COMMENT=) if you want one.

F: “Practice interviewing” may be set in the employee ID file, or at the interviewer ID prompt by adding ,PRACTICE or ,TRAINING after the INTERVIEWER ID.

All other configuration file options are irrelevant in UNIX.

4.1.4 Starting the Interview

You are now ready to start interviewing. The interviewing prompt allows you to interview (I or <Enter>), resume a suspended interview (R) or quit (Q).

If you have allowed practice interviews, you will be in practice mode until you set item F: Do Practice to NO (Standalone; or Quit and get started again by the Supervisor). No data will be saved.

Here is the interviewing prompt:

```
0 Interviews started 0 Completed
Time now (05 SEP 1996 11:57) -- 0:00 since last
interview
Press Enter to Interview or R)esume or Q)uit
-->
```

The information displayed before the prompt can be changed using the INFO_BETWEEN study header option.

You can choose any of these items by entering the first character of the keyword. To begin the interview, press **Enter** and the interview you have specified in the startup configuration will begin.

You may also specify "**VIEW**", "**VIEWONE**", or "**VIEWX**" to view a previously created interview's responses or alter them (see below).

If you are a "debug" mode interviewer ID (D in columns 42-45 of your entry in the employee info file), you may also specify the options necessary to generate a random data file here. This would create data cases with random responses, to check skip patterns and other data entry parameters. If you are doing random data generation, you would specify "**RDG**" or use the RDG command options at this point (See *2.7.2 RANDOM DATA GENERATION*).

If running in a shared files environment, the interviewer can also switch from one questionnaire to another, provided they have a C or D in columns 42-45 of the employee information file.

Entering "**CHI**" will prompt for a <studyname> that will then switch them to the named study.

See *4.2.3 SPECIAL INTERVIEWING COMMANDS. Change Interview* for more information on this.

Also in the shared file environment, interviewers my type "**LUNCH**" or "**BREAK**" to be put in a suspended mode while going on a break or to lunch. This puts a message on the terminal saying they are out, tells the supervisor that they are on a break or at lunch and causes the server log file to mark when the lunch or break time occurred so it can be included in interviewer time reports.

Also see 2*.7.2 RANDOM DATA GENERATION* for information on random data generation and *4.4.2* SURVSUPR MAIN MENU.

4.1.5 Viewing a Previous Interview

In Survent, you may type VIEW at the interview prompt to look a the responses entered in prior interviews and possibly change them. There are also other ways to invoke VIEW mode: run the “survview” program at the operating system prompt, choose the VIEW option in one of the CfMC menu programs, or invoke VIEW mode from the Supervisor program.

View mode allows you to view or edit many cases. If you specify “VIEWX” you can also view or edit many cases, but when you ask to quit out of viewing, your session is terminated instead of putting you back at the <Enter> to Interview prompt. You may also specify “VIEWONE” and it will only allow the viewer to see one interview at a time, and then be quit out of the session.

Using the survent “INIT:” parameter, you can thus control viewing from a menu without the viewers ever getting to an interviewing prompt.

Here is what the screen looks like when you run VIEW mode:

```
Type questionnaire (QFF) file name -->T1798
Type interviewer ID [,special type] or "quit" -->oc
You are Oliver Cook (N/Y)?y
Study name: T1798, comment: 'SEED QUESTIONNAIRE',
length
800 (station 143) -
Looking for CfMC phone-quota-data server:
server found at ldev 2
** SURVVIEW ** for study (T1798) - networked.
Enter password -->
```

Only non-Survent VIEW applications will prompt for a questionnaire name or interviewer ID. If there is a password on the study you will be prompted for the password. The programmer sets the password on the header statement using option PASSWORD=. If there is no password, press **Enter** at the prompt. If you enter the wrong password, you will be reprompted.

You will not see the characters as you enter the password.

Next, you will be prompted for the case ID of the interview you wish to view. If you do not know the case ID to view, you can press **Enter** at the prompt for the case ID, and you will get the first available case. Then the program will display the case number that it has retrieved, the study name, and the interviewer ID of the completing interviewer.

```
Enter case ID to view, Enter for next case, or Q)uit
--> Enter
Got the case. ID: 0001 study code: rrunr intv ID: mary
Press any key to continue
```

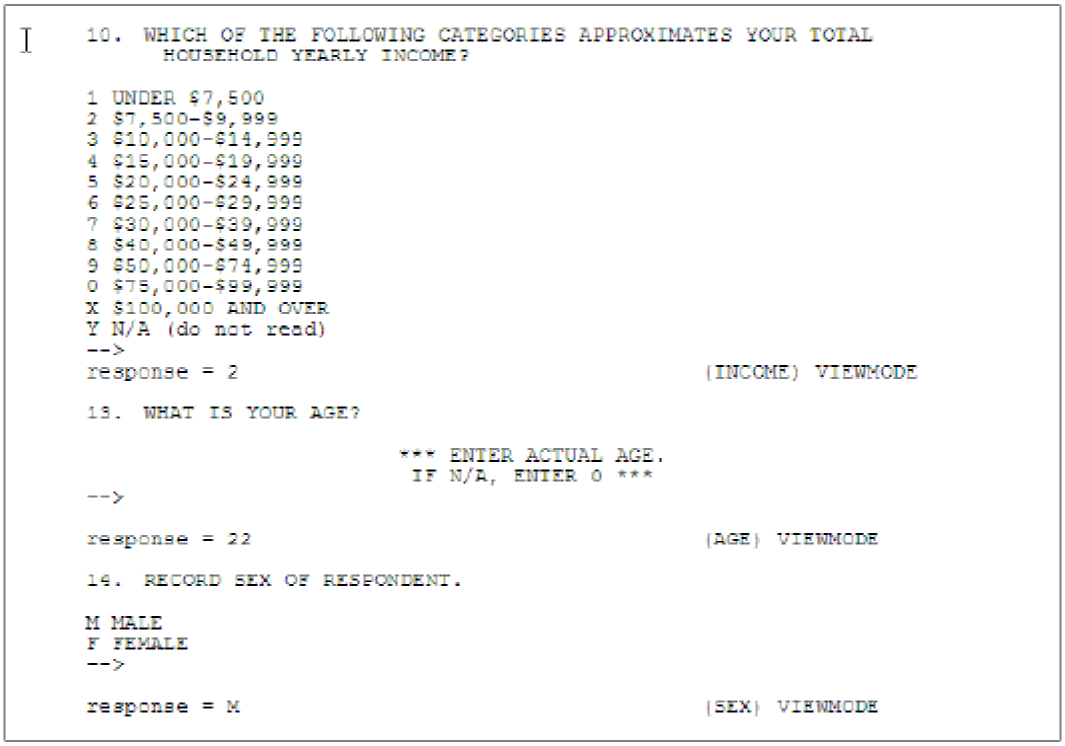
Next you will be prompted for the question(s) you want to view. Enter Question(s) you want -> At this prompt you can say ALL, SAME or enter a list of particular questions you might want to view. The question list can consist of question labels or question numbers, depending on what was used in the specification file. Question numbers should be entered as 1 or 0.1 or 1.55,etc.

Numbers less than one need to have a zero entered before the decimal point (i.e., 0.5, not just .5).

ALL will view all questions answered in the questionnaire SAME will use whatever question list you specified on the previous case question list the questions can be entered one at a time, each on its own line, or several on one line, separated by commas.

```
    EX:
    Enter Question(s) you want ->age
    Enter Question(s) you want ->income
    Enter Question(s) you want ->sex
    Enter Question(s) you want ->Enter
    or
    Enter Question(s) you want ->age, income, sex
    Enter Question(s) you want ->Enter
```

Enter the questions in any order you think of them; the program will show them in the order they appeared in the questionnaire. You will be told if you specify a nonexistent label.



```
10.  WHICH OF THE FOLLOWING CATEGORIES APPROXIMATES YOUR TOTAL
       HOUSEHOLD YEARLY INCOME?

1 UNDER $7,500
2 $7,500-$9,999
3 $10,000-$14,999
4 $15,000-$19,999
5 $20,000-$24,999
6 $25,000-$29,999
7 $30,000-$39,999
8 $40,000-$49,999
9 $50,000-$74,999
0 $75,000-$99,999
X $100,000 AND OVER
Y N/A (do not read)
-->
response = 2                                    (INCOME) VIEWMODE

13.  WHAT IS YOUR AGE?

                        *** ENTER ACTUAL AGE.
                         IF N/A, ENTER 0 ***
-->

response = 22                                   (AGE) VIEWMODE

14.  RECORD SEX OF RESPONDENT.

M MALE
F FEMALE
-->

response = M                                    (SEX) VIEWMODE
```

If you view a question that was not asked in the interview, you will be able to see the question's screen, but the answer will show up as nothing (i.e., no response).

The first question will then appear on the screen—with its response at the bottom of the screen. You will also see VIEWMODE in the lower right corner of the screen, reminding you that you are Viewing, not interviewing.

Questions marked with the compiler commands VIEW, -VIEW, VIEWX, and -VIEWX will affect what you can view. See *3.2.1 INTERVIEW CONTROL COMMANDS*.

VIEWing Data Files with Nonmatching Names

Under Unix, entering <Studyname>,<filename> at the questionnaire filename prompt will let you VIEW data files with names that don't match the study code of the questionnaire.

When starting up Survent, if you say "test,myfile.tr", the program will look for the questionnaire file test^qff and open the data file myfile.tr to be accessed by View mode. You can then display or ALTER answers like you can in any View session.

DOS Survent standalone can get another data file name on the CFG screen.

Commands in VIEW Mode

While in VIEW mode, you can use the following commands:

Enter	moves from one question to another
^	backs up to the previously viewed screen on the current case
ALTER	allows you to change the answer to the current question
ABORT	stops viewing the current case and drops all changes
ENDCASE	stops viewing the current case and save changes up to that point
GOTO	moves around in the questionnaire. Enter * at the prompt to see a list of questions to go to.

Altering Cases

The ALTER command lets you modify the answer to a particular question for a particular case. You invoke ALTER each time you wish to modify a question. The screen will be repainted with that question's screen, and you should enter the new answer. The message at the bottom right corner of your screen will change from VIEWMODE to ALTER to remind you that you are in Alter mode. The original answer to that question will not appear on the screen where you enter the new response, so note the answer before you start ALTERing.

You can also use the “=” character with ALTER to return to the question without making any changes, similar to what you can do with backing up in a questionnaire.

Furthermore, if you have a multiple-response question and you are using ECHOCATS or HIGHLIGHTCATS modes, when you "ALTER" the question, the old responses are left filled in and you can add or delete codes as desired, rather than having to re-enter all the codes. (If you are in the "regular" interviewing mode, it still works as before – you'll need to re-enter all the codes).

When you return from an "Altered" question, you will be returned to that question (instead of the question after it) so you can verify that the change was made.

If you alter the data in such a way that branching is affected, you will not be given any error or warning message, but the rest of the questions you view will be affected. You may be brought down a new path, depending on which questions you asked to view at the beginning. You will not be shown questions from the original first branch and you will not be given any warnings about old data still there. Be careful when altering questions that might affect branching!

Note that you cannot BLANK a question unless the question allows BLANK as a response. Use the CLEANIT utility if you need to blank data that otherwise requires a response. Also, changing a value that changes the order of questions does not blank questions that are no longer answered.

ALTER will require confirmation for the changes when view is done on that case before a case is updated. Altered cases are flagged and may be tracked by MENTOR. At the end of the interview, you will be prompted:

```
This case has been marked as changed.
Update this case (Yes/No)?
```

Enter Y or Yes to update the case with your changes, N or no to remove the changes.

In shared files mode, when you reach the end of the data file, Survent will roll back around in the file automatically, so pay attention when you are letting the program get you the next case. When in standalone mode, you will be told that you're at the end of the data file.

The CfMC Supervisor may also invoke VIEW mode to look at cases.

NOTE: Because VIEW allows modification of data, some shops may not want to have this option. Therefore, you can control use of this feature in the following ways:

1. To disallow “VIEW” entirely, add it to the list of disallowed commands in the suprinit file (“Disallow view”).
2. A password may be placed on the VIEW command by specifying it in the CfMC Parmfile, eg. “Superpasswords: View=Careful”.
3. Passwords may be placed on a particular study to control VIEWing of that study (put PASSWORD= “pass” in the study header).

Moving Around in View Mode

Use Enter to go forward and the Backup key (usually ^) to go backwards in a viewed case. Unlike during an interview, movement by itself does not change any data, for instance, if you ALTER a question and back up over it, it retains its altered response.

Use the GOTO command to skip to a particular question or to another question you want to view. GOTO is sometimes necessary to skip over questionnaire logic that would otherwise cause the VIEW session to end (such as quota checks after the quotas have been filled). Enter the name of the question to go to or enter a label pattern to only see certain questions to choose from. For instance, “Q*” would only show questions that begin with a “Q”. “*” would show all questions.

If you are done making changes and wish to exit the case, enter ENDCASE. You will be prompted as above whether to save the changes or not.

If you are done viewing a case and don’t want to save any changes, type ABORT. If you have used the ALTER command on the current case, changes will be lost. You will be prompted for a new case or to quit.

Or, as another option, you can enter GOTO <label> which will take you directly to that question instead of displaying a list. So, if you enter “GOTO lastquest” on any input line, it will take you to “lastquest” in VIEW mode. If you enter an incorrect label, VIEW will take you to the next question. This is useful in Web mode (webCATI) as well as terminal modeControlling the use of View Mode

Since reviewing data cases and altering responses is something you might not want everyone to do, there are many ways to control the use of VIEW mode. As stated above, each study may have a password placed on it. In addition, here are some other ways to control use of the VIEW feature:

- If an “N” or “O” is placed in the employee file columns 42-45 for a particular interviewer ID, they will not be allowed to use the VIEW feature. If an interviewer is assigned “Practice” mode (P in those columns, or specifying “,Practice” at the interviewer ID prompt), they can view the data but may not ALTER it.
- In addition, you can specify either “D” (Debug mode) or “A” (Alter) to allow an interviewer or supervisor with that ID to modify the data using the VIEW command. If you put {!-allow_alter} in the questionnaire, then no one can alter the data in that portion of the questionnaire.
- To keep supervisors from using the view feature, add the command “Disallow View” in the system’s “suprinit” file (found in the CfMC support directory).
- To put an additional password on VIEW mode for supervisors, add the command “SUPERPASSWORDS: VIEW=pass” in the CfMC “parmfile”.

4.1.6 Resuming a Suspended Interview

If you have suspended any interviews, you will need to resume them to finish the interview. To resume an interview when not using the phone system, enter R (or RESUME). You will be prompted for the name of the resume file. Enter the name assigned to the file when it was suspended (see the SUSPEND command in *4.3 SURVENT COMMANDS*). You will see the comment the interviewer typed in, followed by any questions in a

RESUME block, then return to the question at which the interview was suspended.

Note: If the questionnaire has been extensively changed since you suspended an interview, the interview may not resume. Use the FIXRESUM utility to update the saved information from the suspended interview to match the new questionnaire (See *5.10 FIXRESUM*).

This method of resuming suspended interviews works only for interviewing without a phone file. If the job has a phone file, the interview will automatically be resumed at the appropriate time.

An error message will print if you try to resume an interview collected with a previous version of Survent, or if the case or answer length is different from the current QFF file.

4.1.7 Doing Practice Interviews

Practice interviews do not write a case to disk so data is not saved and no case IDs are assigned; quota and phone files are also not affected. PRACTICE will appear in the lower right corner of the interviewing screen as a reminder to the interviewer. How you invoke practice mode depends upon your interviewing configuration. Refer to one of the sections below for instructions.

In DOS, to practice interviewing you must have set “Do Practice Interviewing” in the configuration file to YES. See *4.1.1 MODIFYING THE CONFIGURATION FILE* for information on the configuration file.

In UNIX, Survent does not have an interviewing configuration file. After starting the Survent program you will be prompted for the study name and then an interviewer ID. Enter an interviewer ID followed by a comma and the keyword PRACTICE (or P or T or TRAINING) to do practice interviews.

```
EX:
Process started.
Type interviewer ID: ALF,practice
```

You can also assign a special interviewer type at this prompt in addition to PRACTICE. Items must be separated by commas.

```
EX:
```

```
Process started.
Type interviewer ID: ALF,practice,special=1
```

This starts interviewer ALF in Practice mode and as a special type 1 interviewer. When interviewers are in “practice” mode, they have the option to NOT rotate questions in rotate blocks. This is so the questions can be reviewed in groups and in sequence when the questionnaire is being reviewed.

There are three ways to control this:

1 To control the default, use the ROTATE_PRACTICE: YES/NO parameter in the CfMC parmfile (in the CONTROL directory). If you say nothing, rotations will occur in practice mode.

```
EX:
ROTATE_IN_PRACTICE: <YES/NO>
```

2 When starting stations in the supervisor, you can use the ROTATE_IN_PRACTICE option to turn on rotations, or -ROTATE_IN_PRACTICE to turn them off.

For example, “START 1-5 BANK,P,-RP” would start interviewers in practice mode and NOT rotate questions.

```
EX:
START #1-5 Bank,Practice,Rotate_in_Practice
```

or

```
Sts #1-5 Bank,P,RP
```

3 At the interviewing station, you can enter ID,-ROTATE_IN_PRACTICE at the interviewer ID prompt, or enter -ROTATE_IN_PRACTICE at the “Between Interviews” prompt.

```
EX:
Mike,-RP
```

Using Practice Mode in Supervised Interviewing

Supervised Survent interviewing does not have a configuration file. Parameters such as practice interviewing may be controlled by the supervisor (SURVSUPR program), the employee info file, or on the interviewer ID line. Interviewers are started on a study from SURVSUPR Main Menu item STS. After specifying the station number and study name, the supervisor can optionally specify interviewing modes. To start a station in Practice mode the supervisor would enter the keyword PRACTICE or TRAINING or a P or T after the study name.

```
EX:
STS 100 example,practice
```

NOTE: A comma must follow the study name. Other mode options are covered in section *4.4.2 SURVSUPR MAIN MENU* under option STS.

4.1.8 Dealing with Abnormal Survent Terminations

Occasionally a Survent interviewing session will be terminated for reasons beyond your control. The reasons for this can be lack of system resources, a hardware crash, a programming error, or a "bug" in the software. There are a few different ways Survent handles this depending on the situation.

Survent "BLOW" errors (controlled termination)

In some cases you will receive a "BLOW" error. This will consist of a message with a short explanation of the problem and an error number. It will also tell you whether it is a "system" problem, a "programming error", or a Survent bug. If it is a "programming error", you need to contact the questionnaire programmer and explain the error; they can fix it. If it is a "system" error you need to check with your system administrator to solve it. If it is a Survent Bug you need to contact CfMC Support (*see APPENDIX G: SURVENT BLOW ERRORS*).

By default, when you receive a "BLOW" error, the interviewer's session will be terminated and any data they have collected saved to a backup datafile in the <study>.b_ directory. You can read the responses in the file saved to determine how the problem occurred, and reenter the responses and SUSPEND the interview to call it back later. We do not automatically allow you continue by default because it is very important that these errors get reported so they can get fixed!

If you have the keyword “**AFTER_BLOW: CONTINUE**” in the CfMC parmfile, the program will allow the session to continue. Please make the interviewers aware that they need to report the problem immediately even though they are allowed to continue.

If you have the keyword “**AFTER:BLOW: SUSPEND**” in the CfMC parmfile, the program will SUSPEND the interview and schedule it for the same time the next day. Again, make sure these errors get reported as they will probably occur again if no action is taken.

Uncontrolled Terminations

There are times when the CfMC programs cannot stop a terminal from getting disconnected or the program from aborting. In these cases one of two things will occur:

- If you have terminals connected to the CfMC server via a “telnet” connection, and the terminal somehow is disconnected or the network connection is terminated, CfMC can sense that the terminal was disconnected; in this case we SUSPEND the interview and reschedule for the next day.
- In other cases the program will just abort. The interview will be lost and you will have to clear the interviewer station and restart it. If this happens at a certain point in the interview more than once, it is probably a CfMC bug and needs to be reported and fixed by us.

4.1.9 Quitting the Program

To exit the program, enter Q (or QUIT) at the between interviews prompt. The data file will be closed, and you will be returned to the operating system prompt. To start interviewing again, simply enter Survent.

4.2 INTERVIEWING WITH SURVENT

Interviewing with Survent is basically self-explanatory — interviewers can follow the directions on their screen. The process is designed to be similar to interviewing with a paper and pencil questionnaire.

Generally, responses are typed at the arrow prompt (-->), **Enter** is pressed and the next question appears on the screen. Responses can be easily changed:

- *before* pressing **Enter**, by backspacing through the response and entering the correct response.
- *after* pressing **Enter**, by using one of the Survent commands explained in the next section to return to the question and re-enter the response.

Survent checks responses to assure their validity. For example, if a numeric response is expected and letters are entered, an error message appears and the interviewer is prompted to press **Enter** and try again.

The responses are never case-sensitive. AA, Aa , aA and aa are all treated the same.

Meta (>) commands and ampersand (&) file references are disabled unless you are in Debug mode.

Other modes of operation will let you mark lists on the screen (Highlightcats mode), do data entry (AUTO_RETURN compiler command), and place multiple questions on the screen at one time (GRID question).

4.2.1 Responding to Different Question Types

Survent expects different responses depending on the type of question that is asked:

FLD (Field) or **CAT** (Category) questions display text on the screen with a list of codes. To respond, enter one or more of the codes exactly as shown, or move to the item and choose it as explained under the HIGHLIGHTCATS feature below.

Multiple responses are entered directly after each other, with or without commas in between. The commas are not put into the data.

If you enter a response not on the code list, the bad response will be identified and you will be prompted to enter a new (set of) response(s). If you enter too many responses, the program will tell you how many are allowed and prompt you to re-enter the responses. If you enter an exclusive response in combination with another response, the program will tell you which one was exclusive and allow you to re-enter your responses.

NUM (Numeric) questions accept numbers and a short list of possible exception codes. Usually the question will have a range of valid numeric responses. Decimal places may be allowed. You need not specify leading zeroes (e.g., 25 is the same as 00025). Commas are allowed as part of the numeric response, eg. 1,000. Negative numbers are specified with a leading dash (e.g., -10), while positive numbers require no sign.

VAR (Variable) questions display underscores (_ _ _) and periods (. . .) on the screen by default. These indicate the minimum number of characters (underscores) and the maximum (periods) that may be entered. You must key in at least the same number of characters as underscores on the response line. The response is typed over these markers. If the response is too long or too short, an error message appears and you are prompted to try again.

TEX (long open-ends) questions allow full-screen editing capabilities. There are 2 modes, BOX mode and LINE mode. BOX mode is used in DOS and Unix systems. The program puts a BOX on the screen and the interviewer types within that box. It will automatically wrap words at the end of each line—and text or lines can be inserted or deleted. Once the text has been entered, **ESC** must be pressed to go to the next question.

Line mode is used in UNIX systems, It prompts for characters every time you press **Enter** and it tells you how many characters are left for the response. The interviewer can type continually at the prompt, letting the words wrap on the screen, or they can press **Enter** at the end of each line, at which point they will be reprompted until they press **Enter** twice.

If you run out of space for TEX question responses on a particular case, Survent displays "No room for text, record by hand." Record the response on paper before continuing on to the next question.

The **EDIT** statement displays a previous TEX or VAR question, and allows changes in the same manner. See *4.2.2 RESPONDING TO SPECIAL QUESTION TYPES, Edit Mode* for more information.

There are two edit modes, BOX mode and LINE mode. DOS uses BOX mode exclusively. The prior text displays on the screen and can be edited using standard DOS movement and editing keys.

UNIX also uses BOX mode editing by default, but it has different editing keys than the DOS version. See *4.2.2 RESPONDING TO SPECIAL QUESTION TYPES, Edit Mode* for information on using the editor for UNIX. UNIX uses Line Mode Editing if you are editing a TEX question that is in Line Mode.

UNIX can optionally use the Line Mode Editor. This displays the text of the question and assigns line numbers, which you choose to edit with editing commands one line at a time. See *4.2.2 RESPONDING TO SPECIAL QUESTION TYPES, Edit Mode* for information on using the line editor.

4.2.2 Responding to Special Question Types

In addition to the response control above, there are some other ways to display and control the response to questions.

Questions Using Autoreturn Mode

Autoreturn mode is used where the questionnaire designer wants the response to cause the program to immediately move on to the next question, without the interviewer pressing **Enter** first. If you see a modified arrow (==>) prompt, you are in Autoreturn mode. When the maximum number of characters is entered as a response, the program will automatically jump to the next question. If you enter less than the maximum and want to go on to the next question, press **Enter** as you normally do.

You cannot use any of the standard interviewer commands at the ==> prompt except for the caret (^). Press **F2** (on a monitor; **Ctrl**-**F**-**2** on terminals) to get a regular --> prompt to enter any other commands.

NOTE: On terminals, you will not see the modified arrow prompt for Autoreturn mode, but the mode will work the same way on all systems.

Highlightcats Mode Response Lists

Highlightcats mode causes response lists from FLD and CAT questions to be displayed in an interactive full screen mode. This is the only way to present a question with more than one screen full of responses, as it lets you move from screen to screen.

Highlightcats mode is often used in DOS which has good screen handling capabilities or for selfadministered questionnaires, where presentation is particularly important.

First, the text and list of responses appears on the screen with the first response highlighted (in inverse video).

On a single-response question, pressing **ESC** or **Enter** at the highlighted response will accept the highlighted response as the answer. To move to another response, enter the response, or use the up or down arrow keys (monitor) or **Ctrl**-**U**,**D**,**L**,**R** (terminal, for up, down, left, or right, TAB will also move down) to position yourself on the item, then press **ESC** or **Enter** to accept the response. You can also program the questionnaire to move to the response whose TEXT matches the input rather than the response code.

On multiple-response questions, you move to the response in the same way as for single-response questions, but then you must mark the response as chosen. To mark it, press the “space bar” or use the + or **INS** (DOS; **Ctrl-P** for UNIX) key. Press **ESC** to accept the question with all of its marked responses. To remove a response after having chosen it, move back to the response on the screen, then use the “space bar” again to toggle it off, or use the “-“ or **DEL** (DOS; **Ctrl-N** for UNIX) key.

To enter keywords on a Highlightcats screen, press the **F2** key (monitor; **Ctrl**-**F**-**2 or Ctrl-** on terminals) to get a prompt, then enter the desired keyword (see *4.3 SURVENT COMMANDS*).

On questions that span more than one page, you can either enter the code or text of the response on the other page, or use the movement keys to move to that page. Use **PgUp** and **PgDown** (or **Ctrl-T Ctrl-V** on terminals) to move from screen to screen, and **Home** (**Ctrl-G** on terminals) or **End** (**Ctrl-E** on terminals) to move to the beginning or end of the list.

Echocats Mode Response Lists

If ECHO_CATS has been specified on the header statement or on a compiler command, when you enter your answer and first press **Enter** on a CAT or FLD question, instead of moving to the next question you will remain on the same question, but all answers entered will be echoed back to you by highlighting their response codes. To add more answers, enter their codes and press **Enter**; you will again see them highlighted. To remove answers, enter their codes preceded by a minus sign (i.e., “-3,5”) and press **Enter**. When finally satisfied, press **Enter** only.

If ECHO_CATS has not been specified, you can put yourself temporarily into Echocats mode on a particular question by typing a period and **Enter** at the question prompt or by adding a period to the end of the response(s) before pressing **Enter**. The response(s) will be highlighted and you can then add or subtract responses as described in the preceding paragraph.

NOTE: ECHO_CATS will override Highlightcats or Autoreturn mode on CAT or FLD questions. Display-only Questions Display questions do not accept any data input, but are used to display messages to the interviewer, or review previous responses. Press **Enter** to continue after reading the screen. You will not see a prompt, but you may enter the caret ("^") or keywords when at one of these questions.

Screen Edit Mode (DOS)

DOS uses the full-screen editor. Use “**Backspace**” to back up one space, “**Tab**” to move to next tab, “**Home**” to front of line, “**End**” to end of line. Use the arrow keys to move up and down.

Toggle the "**Ins**" key to insert or overstrike text. Use "**Del**" to delete text, and "**Esc**" to leave the question.

Screen Edit Mode (UNIX)

This is also a full screen editor, but you need to use Ctrl-keys to move around and alter the text. When entering text, enter your text exactly as you want it to appear on the screen. Words will wrap at the end of the line. Here are the control keys and what they do:

Backspace	Ctrl-H or Backspace key
Beginning of line	Crtl-G
Delete letter	Ctrl-P
Delete line	Ctrl-K
Down arrow	Ctrl-G
End of line	Ctrl-E
Escape	Ctrl-[or ESC key
Insert text	Ctrl-N toggles mode on and off
Left arrow	Ctrl-L
New line	Ctrl-M or Enter key
Right arrow	Ctrl-R
Tab right	Ctrl-I
Up arrow	Ctrl-U

The **Enter** key or **Ctrl-M** in **INSERT** mode will move text down one line. **Ctrl-I** (tab) in **INSERT** mode will move text over one tab stop.

Line Edit Mode (UNIX)

Line mode in Survent allows you to change the contents of a previously asked TEX or VAR type question to which you have done a RETAKE, or caret "^", or used an EDIT question on. The text is displayed on numbered lines, and you commands based on the line numbers displayed.

Here are the valid commands that you can issue:

A# Begins adding lines after line *#.*

A@ Add at end of text. Keep in add mode until user presses **Enter** all by itself.

A< Adds lines before the first line.

A> Adds lines after the last line.

D# Deletes line *#.*

F# Allows intraline editing; identical to “M” command, except that “F” doesn’t strip off line separators.

L# Lists line *#.*

L@ Lists the entire text.

M# Allows intraline editing (within a line); strips off line separators (“\”) before editing, restores them after editing.

M#/ Allows intraline editing from # to the end of file.

CL Cancels all previous commands and restores the text.

E Exits the editor.

H Lists help screen for Edit mode.

U Undelete last line deleted.

If you do not specify a line number in the above commands, the command will affect the current line. Once you have issued an M (or F) command to make changes within a line, the cursor will appear beneath that line. (Doing an EDIT to a VAR question makes the response immediately current for modifying.)

After an M or F command, the following modify commands may be used:

H Gives you Help for Modify mode

D Deletes the character immediately above.

I Inserts a string in front of the character under which the I is located.

X Extends the line by adding a given string to the end of it.

R Replaces characters beginning with the character under which the R is located.

U Removes the effect of the last command.

CL Returns the line to its original form.

. A period will return you to Edit command mode with no changes to the current line. (Used to get out of M#/.)

Enter Ends edit.

You will be asked to verify that you want to end the edit. Answer Y (yes) or N (no).

Note: Line numbers change when you add or delete lines. Do an L (list) command to verify line numbers before using line numbers as part of another command.

For VAR questions, you will be put in "box mode" to edit.

List Edit Mode

On CAT and FLD questions with subtype X, you are shown the current answers on the prompt

(i.e., (0507)-->) and you can use the "+" and "-" in combination with response codes to modify the

answer.

```
EX:
(010406) --> +03-06
```

This will add code 03 to the answer list and remove code 06, leaving codes 01, 03 and 04 as

answers.

Number Line Questions (NUM,R)

This question type presents a number line on the screen below the text as follows:

```
[ ] [ ] [ ] [ ] [ ] [x] [ ] [ ] [ ] [ ] [ ]
```

for 20 or fewer possible entries, OR

```
|-----------------------*-----------------------|
```

for more than 20 entries.

This is presented with the idea of rating something high, low, or neutral, without having to respond with a number. Left is lower, to the right is higher.

The cursor is initially positioned in the middle box or on the asterisk (*). The left and right arrow keys (monitor) or **Ctrl-L**,**R** (terminal) are used to position the cursor. When **ESC** or **Enter** is pressed, an asterisk (*) appears at that position for one second and its position is recorded as the response. Each box or dash (-) is a number above or below zero, with the middle box or asterisk (*) being at 0.

Some feel this creates less bias in a rating scale response than answering with a number in a range. This would only be used for self-administered questionnaires.

Grid Screens

A grid question block is a set of questions that are presented on the screen at the same time to be filled in by the interviewer. It could be used any time a forms type screen is desired, eg:

- Filling in the respondent's name and address
- Having a set of numbers add to 100%
- Gathering name/age, etc. for various household members
- Providing data entry screens
- Filling a tax form, or
- Spreadsheet type applications

While on the grid screen, you can move from question to question by pressing **Enter or Tab** (either before or after a response) to go to the next question, or by using the arrow keys (DOS, UNIX) or the **Ctrl** keys (on terminals in UNIX **Ctrl-U**, **D**, **R**, or **L** for up, down, right, or left, or down). **Home** and **End** are also supported and **Ctrl-G** or **Ctrl-E** (Unix) to move to the top or bottom of the screen. When you get to the bottom of the screen, **Enter** will take you out of the grid if you have met all the grid requirements, otherwise it takes you to the top of the grid. The arrow keys and **Ctrl-U**, **D**, **R** and **L** take you to the top or bottom of the grid when you reach the bottom/top.

When done with the grid screen, press **Esc** to go to the next screen. If all the conditions of the grid are satisfied, you can move forward, otherwise you are returned to the grid screen with the previous answers intact and must continue to change responses until the conditions are met.

If you back up into the grid, the entire answered grid is redisplayed and you are set at the top upper corner of the grid, from where you can continue filling entries. If you back up out of the grid, the entire grid is cleared. The caret (^) key inside of a grid will back you up to the question before the grid. You cannot use it to back up within the grid, use the arrow keys. The caret will back out of the grid from anywhere in the grid.

To refresh the screen in the grid, use **F3** (monitors) or **Ctrl-F-3** (terminals).

4.2.3 Commands at the Interview Prompt

Interviewers have several choices at the Return to Interview or Q)uit prompt. In addition to pressing **Enter** to start an interview, entering **Q** to quit the program, or entering **RDG** to start random data generation with the proper information and/or authorization, they can go on break or lunch, switch to another study, view an aborted interview, conduct an interview with logging taking place after every question and use tracing options.

Going on Break or Lunch

Interviewers may specify that they are on break or at lunch by typing "**BREAK**" or "**LUNCH**" at the prompt. The program will put a message on their screen that they are on break or lunch and it will record how long they have been gone. It will also notify the supervisor that they are on break or lunch.

When the interviewer returns from break or lunch and presses a key, the program will resume in standard interviewing mode.

The server log file records the time when interviewers are on break and lunch, which is reported in the server log reports (see the file lgrptt^spx in the CfMC SURVENT directory). Supervisors can also assign interviewers to break or lunch mode using their "**BREAK**" and "**LUNCH**" commands.

Changing to Other StudiesC

Interviewers can change to other studies without waiting for the supervisor to change them. This is implemented by entering a **C** (or **D** or **B**, which also add other capabilities) in the employee ID file (columns 42-45) for that interviewer ID. If this is set, if the interviewer enters **CHI** at the Return to interview --> prompt, they will be prompted for the study to change to, and attached to the study specified. They will also be prompted for the interviewer ID, giving them a chance to sign on with an ID that has different capabilities (practice mode, special interviewer type, etc.). If the interviewer just presses **Enter** at the interviewer ID prompt after a CHI command, the same interviewer information as last entered will be kept and used.

If you use the CfMC parmfile option "SHOW_STUDIES: YES", when you specify the "CHI" command a list of the available studies is displayed; you can then move around on the list and choose the study you want to switch to (see *4.4.4 SETTING UP THE CONTROL FILES USED BY SURVENT*).

Another feature of the CHI command allows you to control whether interviewers are running with or without he "special interviewer type" mode. The syntax is:

```
CHI <interviewer list> <studyname> <S=# or S=none>
<(-) Dialer> <(-)Practice>
```

Where:

"Interviewer list" can be a list of names or range of numbers; "studyname" can be any 3- to 30- character study name; "S=#" can be "-1" or "none" to turn off special interviewer type, or as many of the numbers (1-9) as special interviewer types assigned; "Dialer" or "-Dialer" is turned on or off if a dialer is used or not; and "Practice" or "-Practice" is turned on or off if in practice mode is in use or not.

Logging Interview Responses

By default, all interviews are LOGGED to an ASCII file that saves responses for a particular interviewer. If your system administrator does not LOG all interviewer responses by default, there are a number of features to turn logging off and on.

First, the interviewer may type LOG at the prompt so that Survent saves all the interviewer responses in an ASCII log file. The log file is named LOG<intv ID> and is stored in the CFMC IPCFILES directory. (See *4.6* for more information.)

Type "-LOG" between interviews to turn off logging.

Supervisors may also put the interviewer in LOG mode when starting interviewers by using their own LOG command.

They would enter:

```
          START <interviewer list> <study>,LOG
```

This is used to log the interviewer responses, instead of having to put a LOG command in the study header. If the LOGGING keyword is included in the study header, then responses will be logged in any case.

To turn off logging, the supervisor can use:

```
     STOPLOG<interviewer list>
```

To turn logging back on, use "LOG <interviewer list>". The interviewer list must be active interviewers for the LOG and STOPLOG commands.

If you NEVER want logging, the command "INTERVIEWER_LOGGING=No" in the questionnaire header will disallow all logging of interviewers on the study.

Also, if using INTERVIEWER_LOGGING=AFTER_EVERY_QUESTION in the questionnaire header, the program will write the logged responses AFTER EVERY QUESTION instead of at the end of the interview. Be careful when using this because it can cause a lot of overhead on your computer.

Setting Capabilities at the interviewer ID prompt

When prompted for an interviewer ID, interviewers typically just enter an ID as it is in the Employee information file. If desired, interviewers can also add a command at the end of the ID to put themselves into a particular mode. The modes include practice (or training), special interviewer type and whether to be attached to a dialer. This is in addition to the Supervisor starting them up with those same capabilities.

To start up in Practice or Training mode, add ",P" or ",T" after the interviewer ID:

```
EX:
MJB,P
```

To start up as a special type interviewer (that the interviewer's normal interviewer ID doesn't include), add ",S=#" after the interviewer ID. NOTE: If the Supervisor also issues an S= on the Start command, that will override this setting.

```
EX:
MJB,S=2
```

To start up attached to a dialer (EIS or MSG PRO-T-S) when running under a server, add ",DIALER" or ",ATMS" after the interviewer ID.

```
EX:
MJB,DIALER
```

To switch to NOT running with a dialer, add “, -DIALER” or “, -ATMS”.

To start up as a validator, enter a “V” or “VALIDATOR”. This will assign you as a person who will receive the phone record and data record after an interviewer completes a call to get additional information or verify information. Supervisors may start interviewers as validators using the STS command “STS <interviewers> <study>,V”.

```
EX:
MJB,V
```

To log on with a different terminal type, use T=## (see terminal list above)

```
EX:
MJB,T=11
```

4.3 SURVENT COMMANDS

These commands allow you to go back to previous questions, review responses, or do other interview controlling actions. Interviewers should be trained to use the commands you feel they will find necessary to function comfortably in the program. Most commands can be disallowed programmatically.

These commands are reserved words in Survent. They may not be used as responses or question labels. However, variations on these words are permissible. RESET, for example, is not an acceptable response; RESETTABLE is.

At the standard SURVENT prompt (-->), or after pressing **F2** (on a monitor; **Ctrl**-**F**-**2 or Ctrl-\** on terminals) on the special type questions (Highlightcats or Autoreturn mode, Number Line questions or Display-only questions) you may enter the following Survent commands. For information on how to program keywords into function keys, look in the INDEX file in \CFMC\SURVENT for an example of how to do this.

When you have already entered text for a question and want to stop to enter a command keyword, use the **F2** key to do so (monitors; terminals use **Ctrl-F-2**). In most other situations, you can enter the keyword at the question prompt or as the first thing in a TEX box.

4.3.1 Standard Interviewing Commands

These commands are allowed in almost all questionnaires.

Backing up (^ And =) Entering ^ (caret; on most keyboards, **Shift**-**6**) and then pressing **Enter** instead of a response returns the interviewer to the previous question. Survent backs up one question when a ^ or < are entered. As the interviewer backs up, the responses are displayed at the bottom of the screen. Backing up requires re-entering all responses forward to the point where the first ^ or < was entered unless you enter the equal sign (=) as you scroll back down.

Entering = retains the original response. If at any question you have backed up to you enter a response other than =, it will blank all responses from there down to the point where you started backing up and you will have to re-enter them.

If you are in Echocats mode, when you enter =, it marks the relevant codes that were previously answered and then allows you to continue adding or removing codes.

If you are in Highlightcats mode, it automatically remembers the prior responses when moving forward.

You may specify some other string besides the caret "^" to be used to back up with in the PARMFILE by specifying "BACKUP_CMD: xxx". If xxx=BBB, then interviewers could type BBB<Enter> whenever they wish to back up at a data entry prompt or display question.

However, the only key besides "^" that can back up on a Highlightcats Screen is the "<". If you say "BACKUP_CMD: <", the "<" will be allowed on all screens just like the "^".

The ability to use this command can be turned on and off during a questionnaire by using the compiler commands ALLOW_BACKUP and -ALLOW_BACKUP.

In addition, the programmer can control how many consecutive backups can be made my using the command !maximum_consecutive_backups=#. (See, *Chapter 3, 3.2.1 Interview Control Commands,* for more information.). This is to prevent interviewers from having to reenter intermediate responses.

HELP (supervised only)

If an interviewer wants to get the supervisor's attention, enter HELP. This will send a message to the supervisor if the interviewing is being controlled by SURVSUPR. "HELP" cannot be disallowed.

SAVEDATANOW

This command is seldom used, but it saves the data in case the interviewer loses contact with the CfMC server due to a network or server crash, so it can be fairly important; it saves the data in the local directory with a random name similar to "blow" file names. "SAVEDATANOW" cannot be disallowed.

SHOW

The SHOW command allows an interviewer to display previous responses. Typing SHOW instead of a response causes a prompt for a question number or label, at which point there are three options. The interviewer may enter:

- · A single question number or label to start the display at that question.
- · **Enter** or SAME to keep the interview at the current question.
- · Wild card characters as described in RESET.

Here is a sample SHOW prompt:

```
EX:
Show question (label or pattern) --> Q5
```

The text screen and response from question Q5 would be displayed. The interviewer would press the plus key (**+**) to see question 6, then press **+** again to see question 7, etc. To go backwards, press the minus key (**-**). Pressing **Enter** at any point returns you to the question where you originally entered SHOW.

There are some messages to ensure that interviewers enter the proper information. After entering SHOW, if an interviewer enters a question number or label that has not been asked, this message is displayed:

```
That question exists, but was not asked,
Enter to continue.
```

If a nonexistent question number is entered then this message is displayed:

```
Label ( ) not known, Enter to continue
```

You can only SHOW a question that was previously asked. If you want to enter a question number, you can enter it as QQ# (i.e.: QQ15) or as just the number (i.e.: 15). If the number is less than one, it must be preceded by a zero (i.e.: 0.7, not .7)

The "SHOW" command cannot be disallowed.

4.3.2 Often Allowed Interviewer Commands

These commands are used if the program allows SUSPENDing interviews to be finished later, or includes a SPECIAL block for comments or help or a TERMINATE section to deal with respondents who cannot finish the interview.

REDIAL

If an interviewer is using a dialer and the phone line is disconnected for any reason, the interviewer can cause the dialer to redial the number by using the REDIAL command. This sends a message to the dialer to redial the current phone number at that extension. No history of this action is kept, other than the redial command is logged in the Server log files.

SPECIAL

If a SPECIAL block has been set up for the study, at any point in the questionnaire the interviewer may enter SPECIAL to go to that block and execute the questions there. Afterwards, you will be returned to the question you were on when you entered SPECIAL. This is typically used to type comments or to get help.

You cannot type SPECIAL, SUSPEND or TERMINATE inside the Special block.

The ability to use this command can be turned on and off during a questionnaire by using the compiler commands ALLOW_SPECIAL and -ALLOW_SPECIAL.

Furthermore, there is another option. It is SPECIAL_CMD:<keyword>. This command allows the interviewer to type a keyword to get to the SPECIAL block. The new parameter file SPECIAL_CMD:XX lets you assign some keyword in addition to “SPECIAL” to go to the SPECIAL block.

You could, for example, assign a keyword (such as “HELP”) if you use the special block as a help block. Or, you could use “Comment” if you would like it to record comments. Another way you can utilize this is for shorter words (or company codes), such as ZZ, etc.

NOTE: Whatever keyword you assign, if a text answer matches that keyword, it will attempt to move it to the SPECIAL block.

SUSPEND

Interviewers will sometimes have to stop interviews they hope to have completed at a later time. The SUSPEND command stops an interview and saves all the responses for a later date or time.

Entering SUSPEND instead of a response causes Survent to stop the interview and create a file in which to save responses. In Survent without the phone system, this file will be given a random name by default, which assures you get a unique name.

You can accept this name or provide one of your own. We suggest you provide your own. If you provide a name, make sure it follows operating system naming conventions. Duplicate names are not allowed – if you enter a name that already exists, you will be told to enter another name. Record the file name assigned or chosen, as it will be needed to resume the interview later (see *4.1.6 RESUMING A SUSPENDED INTERVIEW*).

If the questionnaire has a SUSPEND block, these questions are asked before the interview ends. Additionally, the interviewer is asked to enter optional informational text that will be displayed again when the interview is RESUMEd. The maximum number of characters is 79.

```
Enter a comment (max 1 line) to save with
suspended interview
-->
```

You cannot type SUSPEND during SHOW or RETAKE, or in the SPECIAL or AFTER_QUIT blocks, as these are not in the normal questionnaire flow. Once back in the normal flow, the interview can be SUSPENDed.

Suspended cases will not be assigned a case ID until they are resumed and completed, unless you have an SPC,A or SPC,D prior to the suspend.

If you forget suspend information, the program SUSPRES will create a file which shows you the filename, interviewer ID, when suspended and the answers collected up to the suspend point. It will also create a data record you can append to your completed dataset if you'd like. (See *5.12 SUSPRES* for more information on this program.)

The ability to use this command can be turned on and off during a questionnaire by using the compiler commands ALLOW_SUSPEND and -ALLOW_SUSPEND. You also cannot suspend prior to a !PHONE,9 statement when using the phone system.

Note: If the CFMCRESUME variable is set, the resume files will be written to a subdirectory off of the RESUME variable named <study>.R_. If no CFMCRESUME variable is set, the resume files will be written to a subdirectory off of the local directory named <study>.R_.

Phone System Note: Jobs using the phone system will automatically name the suspend records with a random unique name and bring them up at the appropriate time. See *6.5 INTERVIEWING WITH THE PHONE SYSTEM* for more information. You are not given the option to specify a different name.

TERMINATE

Interviewers may need to stop interviews that will not be completed, but save the responses generated so far. Additionally, you may want to have more control (than ABORT) over what happens when an interview stops. The TERMINATE command allows both these things. Entering TERMINATE instead of a response causes a skip forward to a question with a label of TERMINAT, usually at the end of the interview. If there is no such label in the questionnaire from the point where the command is given, the program remains at the current question.

The interview then does whatever has been specified from the TERMINAT label on. There can be multiple Terminate blocks. Each will have a unique name (TERMIN_A, TERMIN_B, etc.), but the interviewer always enters TERMINATE. The program will go to the next Terminate block. See *2.3.5 QUESTION LABEL LINE, Question Label* for more information.

The ability to use this command can be turned on and off during a questionnaire by using the compiler commands ALLOW_TERMINATE and -ALLOW_TERMINATE.

NOTE: The command can be abbreviated to TERM.

4.3.3 Advanced Interviewing Commands

These commands are typically not allowed, but may be allowed if the interviewers have a good understanding of the questionnaire and want to be able to move around in the survey or drop out of unfinished surveys.

ABORT

Interviewers also may need to end interviews that will not be completed without saving the data. Entering ABORT instead of a response causes the interview to end with no record of it saved in the data file.

The ability to use this command can be turned on and off with the ALLOW_ABORT compiler command. Interviewers with debug capability (using interviewer IDs BKOK or DBUG or having D or B assigned to you in the employee file) can use this command anytime.

RESET

An interviewer can go directly to a previous question with the RESET command. The RESET command can back up over many questions, unlike the ^ (caret) command that can back up only one question at a time. This command also deletes responses as it backs up. Entering RESET instead of a response causes a prompt for a question number or label, at which point the interviewer can enter the number or label of a question that has already been asked.

```
EX:
Reset to question label (* for list) --> Q4.A
```

Question Q4.A would appear on the screen along with its original response. This question, and all subsequent questions to the point where the interview was reset from, will have to be reasked unless you use =, as described in the ^ command above.

If the question is a TEX question, the interviewer has the option of adding to the existing response or deleting the response and entering a new one by using full-screen Edit mode for monitors, or the line editor for terminals (see *4.2.2 RESPONDING TO SPECIAL QUESTION TYPES,* Edit Mode).

You can only RESET to a question that was previously asked. If you want to enter a question number, you can enter it as QQ# (i.e., QQ15) or as just the number (i.e., 15). If the number is less than one, it must be preceded by a zero (i.e., 0.7 not .7).

At the prompt, in addition to specifying a question number or label, you can enter a question label subset reference, such as *, # or ?. Survent will then display all of the labels (prior to the current) with that reference and let you move around on the screen and pick the one you want to go to. If you say *, Survent will show up to the last 200 question labels that were executed. If you say S*, it shows you all the labels that begin with S. The special characters ? and # mean one letter or number respectively. Here are some examples:

S??### Questions beginning with S, followed by two letters and then three numbers.

AB*# Questions starting with AB, then anything else, then one number.

???? Questions that have four letters.

Note: A RESET command may be overridden if the interviewer decides that there is no need to reset. Enter SAME at the prompt of the question you want to RESET to, and you will keep the interview at the current question.

The ability to use this command can be turned on and off during a questionnaire by using the compiler commands ALLOW_RESET and -ALLOW_RESET.

RETAKE

The RETAKE command is useful with questionnaires that utilize branching. This is when questions' responses determine which subsequent questions will be asked. RETAKE checks the interview for any necessary changes in branching due to the changed response.

Entering RETAKE instead of a response causes a prompt for a previously asked question number or label.

For example:

```
EX:
Retake to question (label or pattern) --> Q6
```

Question Q6 would appear on the screen along with its original response. If the question is a TEX question, the interviewer has the option of adding to the existing response or deleting the response and entering a new one by using full-screen or line Edit mode.

After the change is made (or you use =, as described in the ^ command earlier), the program takes a moment to move through the questionnaire, looking for branching discrepancies; if a discrepancy is found, a new response is requested. This question, and all subsequent questions from that point on will have to be asked.

RETAKE will usually protect you from collecting bad data; however, there is the possibility that a changed answer didn't affect the branching structure, and (because of its content) should cause some questions to be reasked.

You may use the same wild card characters (*, #, ?) as described in RESET.

NOTE: A RETAKE command may be overridden if the interviewer decides that there is no need to retake. Entering SAME at the prompt of the question to RETAKE and you will keep the interview at the current question.

The RETAKE command works only if the questionnaire writer has allowed it by using the ALLOW_RETAKE compiler command.

You can only RETAKE to a question that was previously asked. If you want to enter a question number, you can enter it as QQ# (i.e., QQ15) or as just the number (i.e., 15). If the number is less than one, it must be preceded by a zero (i.e., 0.7 not .7).

4.3.4 Test/Debug Mode Interviewer Commands

These commands typically are used only by testers or questionnaire debuggers.

ABORTOS (DOS)

When doing continuous interviewing (option 7 on the configuration menu), interviewers may need to stop interviewing and return to the operating system (OS). Entering ABORTOS instead of a response causes the interview to end, with no record of it in the data file, and returns the interviewer to the OS prompt.

The ABORTOS command will work even if ALLOW_ABORT is not specified as an option in the questionnaire's study header; however, you must have allowed continuous interviewing in your configuration file (see *4.1.1 MODIFYING THE CONFIGURATION FILE*).

ENABLE_DEBUG

This allows you to have Debug mode capability even if you are not signed on with a Debug mode interviewer ID.

FORCED_ABORT

Allows the tester to abort the interview even if ABORT is not allowed and you are not a Debug mode interviewer ID.

GOTO

If you are a “debug” mode interviewer you can use this to move forward in the questionnaire. This allows you to test sections later in the survey without answering intermediate questions. Of course, you need to be careful that later questions do not require answers to questions you have skipped.

This is also very useful in VIEW mode to go to the item you want to view.

GOTO can be used to go backwards or forwards. You are prompted for the question to move to. You may enter a label or use the same wild-card characters as described under RESET.

4.4 INTERVIEWING CONFIGURATIONS

This section will cover the various possible configurations for interviewing. Also see Appendix G: HARDWARE AND SYSTEM CHECKLIST for information about setting up a networked system and setting up the SURVSUPR and the SERVER.

The three basic interviewing configurations are described below. Some configurations are system dependent and therefore may not be appropriate for your particular interviewing operation, hardware or operating system.

Standalone Interviewing

Each interviewer has their own phone file, data file, questionnaire file, and quota file. This configuration is recommended only for testing or non-connected interviewing stations. You cannot run standalone and networked survent on the same set of files at the same time. Standalone Survent saves the data case and closes the file after every interview.

DOS Survent will bring up a configuration menu you must fill in. It does not use the CfMC EMPLOYEE information file.

UNIX Survent has no configuration screen. They use the EMPLOYEE info file and TTYINFO file to get all device and interviewer information. All you do is specify the name of the questionnaire you will be using and interviewer ID. This allows you to emulate running under the CfMC server.

Shared Files Interviewing

Using the CfMC study file server (SERVER) to facilitate multiple interviewers working on the same study and therefore same study files (data, phone, resume, quota). Data from all SHARED FILES INTERVIEWS will be collected in one data file, saving you the trouble of concatenating data files. Phone files and quota files are also maintained centrally. To run in a shared files mode, run NETSURV rather than Survent, after making sure that all needed files are put where they would be needed as if running in Supervised mode.

Supervised Interviewing

Multiple interviewers using the CfMC study file server (SERVER) in order to share study files while having the SURVSUPR program supervise the interviewing. Adding the supervisory program (SURVSUPR) to the SHARED FILES INTERVIEWING environment offers monitoring of whole studies and individual interviewers, and control of quotas and phone parameters and information for the study. You may supervise all jobs with one supervisor or split them between supervisors.

4.4.1 SURVSUPR, SERVER and Survent Overview

SURVSUPR is a control program for Survent interviewing. When it is used, one or more interviewing stations can be controlled by the SURVSUPR station. The SURVSUPR station can check the status of any or all supervised interview processes, and visually monitor the progress of interviews. If desired, several different questionnaires may be started and controlled by the same SURVSUPR. Several SURVSUPR programs, independent of each other, can run simultaneously on the same system. You can also have shared files interviewing (netsurv) going on at the same time.

There are several advantages, however, to using the SURVSUPR:

- It assigns interviewers to particular studies instead of relying on the interviewer to choose the right one.
- Supervisors can watch and control the filling of quotas and the progress of the study for its markets.
- Interviewer statistics help manage how many are needed and who needs help.
- It allows the SURVSUPR station to monitor the progress of each interviewer.
- Supervisors can VIEW collected cases and/or make changes.

In effect, the SURVSUPR allows more efficient management of the interviewing process.

The CfMC SERVER is a continually running communications program that communicates with all the processes involved in the active studies. The SERVER requires no user interaction once it has been started. It is required for any sharing of study files, i.e., multiple interviewers working on the same study.

Survent is the program used by the individual interviewers to administer the interviews. Survent can either be run STANDALONE (without a SERVER), and therefore exclusively own all local study files or SHARED FILES (with a SERVER) and share study files with other Survent interviewing processes.

LIMITS: One SERVER can have no more than 1000 active studies at one time. Operating system limitations on the number of files open may make this smaller on some systems. There may be no more than 250 interviewers under one server in DOS, and 10000 in UNIX (operating systems may further limit this number).

Typical Steps in Running a Job

1 Start SERVER

2 Start SURVSUPR

In SURVSUPR:

- **STS** - Start Stations
- **QSS, SPI, MON, MARKET** - Check quotas or markets, get status info, monitor interviewers
- **CHI, VIEW** - Change interviewer to other studies, view completed cases
- **STP** - Stop stations
- **SDS** - Shut down studies

3 SERVER:DOWN - Shut down SERVER

4 QUIT - Quit Supervisor

NOTE: The SERVER program may be kept running and only taken down periodically. You can do backups while the SERVER is running, but any files the SERVER owns will not be backed up – this includes IPC files, the current LOG file, and any study file that is not shut down.

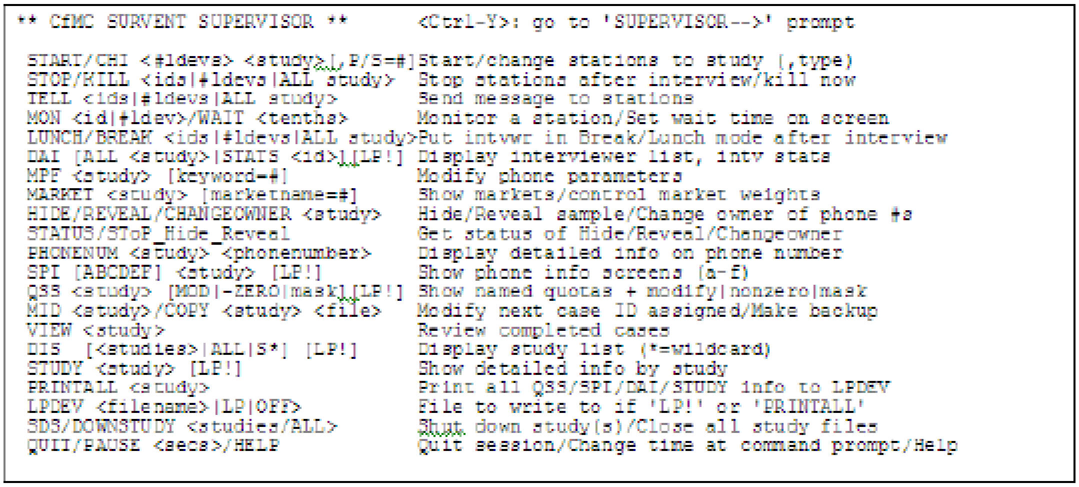
4.4.2 SURVSUPR (SUPERVISOR) Commands

This section explains all of the options available in SURVSUPR. To read a suggested approach to starting up a new job, see *Appendix G: HARDWARE AND SYSTEM CHECKLIST*.

To run SURVSUPR, enter:

```
SUPER
```

Here is the main menu. This menu will not be displayed every time the screen is repainted. The menu is displayed when SURVSUPR is first executed and subsequently when HELP is entered at the command prompt:



```
** CfMC SURVENT SUPERVISOR **      <Ctrl-Y>: go to 'SUPERVISOR-->' prompt

 START/CHI <#ldevs> <study>[,P/S=#]Start/change stations to study [,type)
 STOP/KILL <ids|#ldevs|ALL study>  Stop stations after interview/kill now
 TELL <ids|#ldevs|ALL study>       Send message to stations
 MON <id|#ldev>/WAIT <tenths>      Monitor a station/Set wait time on screen
 LUNCH/BREAK <ids|#ldevs|ALL study>Put intvwr in Break/Lunch mode after interview
 DAI [ALL <study>|STATS <id>][LP!] Display interviewer list, intv stats
 MPF <study> [keyword=#]           Modify phone parameters
 MARKET <study> [marketname=#]     Show markets/control market weights
 HIDE/REVEAL/CHANGEOWNER <study>   Hide/Reveal sample/Change owner of phone #s
 STATUS/SToP_Hide_Reveal           Get status of Hide/Reveal/Changeowner
 PHONENUM <study> <phonenumber>    Display detailed info on phone number
 SPI [ABCDEF] <study> [LP!]        Show phone info screens (a-f)
 QSS <study> [MOD|-ZERO|mask][LP!] Show named quotas + modify|nonzero|mask
 MID <study>/COPY <study> <file>   Modify next case ID assigned/Make backup
 VIEW <study>                      Review completed cases
 DIS  [<studies>|ALL|S*] [LP!]     Display study list (*=wildcard)
 STUDY <study> [LP!]               Show detailed info by study
 PRINTALL <study>                  Print all QSS/SPI/DAI/STUDY info to LPDEV
 LPDEV <filename>|LP|OFF>          File to write to if 'LP!' or 'PRINTALL'
 SDS/DOWNSTUDY <studies/ALL>       Shut down study(s)/Close all study files
 QUIT/PAUSE <secs>/HELP            Quit session/Change time at command prompt/Help
```

There is no limit to the number of SURVSUPR sessions running at the same time. Each supervisor "owns" any stations they start, but otherwise all study information and changes can be done from any supervisor station.

SURVSUPR is in one of two modes at all times — Run mode or Command mode. You must be in Command mode to issue any commands; use Ctrl-Y or your assigned system BREAK key to get the command prompt.

All commands typed by supervisors are logged to the server LL log files. Also, changes to quotas and phone parameters are logged, so you can retrace what happened during a shift if there is a problem. See *4.4.3 CfMC SERVER* for more information on the Server Log files.

Run Mode

Dots scrolling across the bottom of the SURVSUPR screen (or a whirligig design spinning) indicate SURVSUPR is in Run mode. While in Run mode, SURVSUPR is communicating with the SERVER and the interviewing stations. The speed of the dots or spinning indicates the load on the network, i.e., the number of active stations signed on. When the activity slows down, this indicates that SURVSUPR is busy handling the activity of the stations.

Pressing **Enter** at the command prompt or **ESC** on the interactive screens will return the system to Run mode. If you don't return to Run mode, the system will return to Run mode after 120 seconds (or the time specified in your PAUSE command) of inactivity. It is best to be in Run mode when not actively giving commands. Staying at the command prompt may cause messages from other processes to queue up.

Command Mode

Pressing **Ctrl-Y** or your system break key (UNIX) while in Run mode will put SURVSUPR into Command mode. When in Command mode, Run mode operations are put on hold and SURVSUPR looks to the command prompt for user input. If you don't return to Run mode, SURVSUPR will return to Run mode after 120 seconds (or the time specified in your PAUSE command). Pressing **Enter** at any prompt will return you to Run mode.

NOTE: After a command, you will be returned to command mode for another 120 seconds expecting another command unless the "SUPER_REPROMPT_ON_COMMAND" parmfile parameter is set to

“NO” (see 4.4.4 SETTING UP SURVENT CONTROL FILES). To change the program to always go into “run” mode, add SUPER_REPROMPT_ON_COMMAND: NO to the parmfile.

When entering a command, if the option requires a study to be specified, enter it after the command or press <Enter> after the command to be prompted for the study.

```
EX:
DIS JB123
```

If the command requires an interviewing station or stations to be specified, enter it after the command and use the following syntax:

```
#1,3,5 do the command on stations 1,3, and 5
#1,10-13 do the command on stations 1 and 10 through 13
```

NOTE: If you enter the command with no study or station(s), you will be prompted for the missing information. You can specify either stations or interviewer IDs in many places. Since both of these can be numeric, you put a # before the station number to tell them apart.

```
EX:
STP #27-29
```

Signing off

The supervisor cannot quit (QUI) until all interviewer stations started by it show an inactive status (not RUNNING, MONITOR, ACTIVE, or ENDEDINT; see DAI for description of statuses), indicating that the station was either terminated by the supervisor or that the interviewer quit and exited to the operating system.

Using Command Files

Commands can be executed via a pre-written command file. The command file is invoked with an ampersand (&) followed directly by the command file name, for example:

```
Enter a SURVSUPR command-->&start.spx
```

To execute the command file without seeing the command lines, say “&-start.spx”.

You can also set up a defined keyword to execute a command file or set of commands. In this case, the supervisor would type and AT sign (@) to invoke the defined command:

```
    Enter a SURVSUPR command-->@start
```

Using Meta Commands

You can use “meta” commands to do operating system functions while in Command mode. Just preface any command or program you want to run with a “>” and it will do the function. You can also do some of these functions from the CfMCMenu or SuperMenu program. All of these commands of course are also available at the operating system prompt. Here are some commands you may find useful:

Function	**UNIX**	**DOS**
List questionnaire files	>ls –l ../*.qff	>dir ..*.qff
List data files	>ls –l ../data/*.tr	>dir ..\data*.tr
Edit/view a file	>vi <filename>	>edit <filename>
Copy a file	>cp <file1> <file2>	>copy <file1> <file2>
Run mentor to do reports	>mentor spec –list	>mentor spec –list

You can use full filename conventions to access files in other directories or groups on your computer. You can, of course, run the editor of your choice to edit files. When running Mentor, “spec” refers to the specifications file and “list” to the listed output file of your choice.

Receiving Messages from Interviewer Stations and the Server

When in Run Mode, the supervisor will see any interviewer BLOW messages or OUT OF NUMBER messages at their screen. If the interviewer types “HELP”, a help request message will appear. There are also messages when stations are started and stopped or for other system activity.

You can have the programmer set !SPC,E statements throughout the questionnaire and the supervisor will be able to see where the interviewer has interviewed up to based on the messages, either on the run mode screen or in the DAI interviewer activity screen and monitored interviewer list, depending on the option you choose.

Supervisor Commands by Function

The supervisor has many commands, some of which are not displayed on the default HELP screen. Here is a list of some of the commands sorted by function. Commands are also sorted from most used to least used.

Italicized commands (below) are the ones displayed on the help screen. The others are not.

The commands are shown in their longhand and shorthand versions, you may use either.

Supervisor Functional Commands

Command	**Function**
Ctrl-Y	Get command prompt
HELP	Show help screen
QUIT	Quit the supervisor program
SERVER:DOWN	Shut down server before quitting
SERVER:DOWN_NOW	Shut down server even if studies are live

Interviewer Control

Command	**Comm and (shorth and)**	**Function**
START_STATIONS	STS or START	Start interviewers on study
CHANGE_INTERVIEWERS	CHI	Change interviewers to new study

GATHER		Attach any orphaned stations to super
BREAK		Put interviewers in “break” mode
LUNCH		Put interviewers in “lunch” mode
STOP	STP	Stop own interviewers after current interview
KILL	KIL	Stop own interviewers at next keystroke
CLEAR		Stop own stations immediately
SERVER:CLEAR		Totally remove any station from server
SERVER:CLEAR_EXTENSION ##		Clears stations by their phone extension instead of device number

Interviewer Related

Command	**Command (shorthand)**	**Function**
DISPLAY_ACTIVE_INTERVIEWERS	DAI	Display interviewer activity
MONITOR	*MON*	Monitor interterviewer screens live
TELL		Send message to stations or ALL

Quota Related

Command	**Command (shorthand)**	**Function**
QUOTA_SHOW_SCREEN	QSS	Interactive show/modify quota screen

MODIFY_QUOTA	MQ	Command line Modify quotas
NUMBERED_QUOTA_SHOW	NQS	Show/modify numbered quotas
NUMBERED_QUOTA	NQ	Command line mode numbered quotas

Sample Control

Command	Command (shorthand)	Function
MODIFY_PHONE_FILE	MPF	Modify phone parameters
HIDE	HID	Hide phone numbers by criteria
REVEAL	REV	Reveal numbers previously hidden
MARKET	MKT	Show #s by market/modify weights
CHANGE_OWNER	CHO	Change owner of phone numbers
HIDE_REVEAL_STATUS	HRSTATUS	Say status of hide/reveal/cho operation
STOP_HIDE_REVEAL	STPHIDRE	Stop currenthide/reveal/ cho operation

Sample Informational

Command	Command (shorthand)	Function
SHOW_PHONE_INFO	SPI	Show summary phone info screens
PHONE_NUMBER	*PHN or PHONENUM*	Show all info on this phone number
LIST_PHONE_RECORDS	LIST	Shows info on selected sample records

SHOW_STACK	SS	SS Show next available phone number in stack

Study Control

Command	**Command (shorthand)**	**Function**
SHUTDOWN_STUDY	SDS	Shut down inactive studies
DOWNSTUDY	DOWN	Shut study files even if others are open
SERVER:CLEARSTUDY		Totally remove study from server
LOAD_STUDY	LOAD	Load a study under supervisor

Study Informational

Command	**Command (shorthand)**	**Function**
DISPLAY_STUDY_INFO	DIS	Display summary info by study
STUDY		Show complete study information
PRINTALL		Send all info on study to LPDEV
LPDEV		Change/close file to write listings to

Dialer-related commands

Command	**Command (shorthand)**	**Function**
ATMS	ATM	Start interviewers under a dialer
ATMS_VALIDATE	ATMV	Start up interviewer as “Validator”

DIALER_STATUS	DIALSTAT	Tell which dialers are up
DIALER_SUMMARY	SUMMARY	Tell # on each study/dialer/waiting
SERVER:CLEARSTUDYNOW		Clear study from dialer
SERVER:DIALER:WIPEOUT		Clear dialer from system

<u>**Data Related**</u>

Command	**Command (shorthand)**	**Function**
MODIFY_CASEID	MODID	Show/modify next case ID assigned
VIEW		View completed cases
COPY	COP	Copy data file to new name
SERVER:FASTCOPY	SUMMARY	Copy big file to new name faster

SUPERVISOR FUNCTIONAL COMMANDS

Ctrl-Y Get command prompt so you can type other commands

This returns the supervisor from "run mode" to "command mode". Use the "Pause" command to control how long to stay at the prompt between commands (2 minutes by default). Pressing "Enter" returns you to "run" mode.

ALLOW_KILL_SURVENT: ALL/NOBOSS

Allows Supervisors to kill other supervisors' started Survents. ALLOW_KILL_SURVENT: All lets you kill any Survent session on the system, and ALLOW_KILL_SURVENT: NOBOSS allows you to kill independent Survents (such as webSurvents or webCATI sessions), but not interviewers started under some other supervisor.

If you can specify this, you can also “STOP” interviewers that you don’t own. In particular, you can stop webCATI interviewers (that aren’t controled by any supervisor).

BACKSTART (HP-UX)

This command replaces START to start stations on HP-UX terminal controllers.

HELP Display the help menu

This is a list of the most often used options. If you have options that you want to include in the HELP display, you can alter the HELP screen in the CfMC msgfile.raw file and remake the system message file using the MAKEMSG program.

ONLY_BOSS_CLOSES_STUDIES:yes

In the parmfile, this overrides the server's default behavior of closing opened studies after X number of minutes. If this is set, the only way to close a study is by using the following from the supervisor.:

```
sds <study>
down <study>
```

or

```
server:clearstudy <study>
```

QUIT Quit the program

The supervisor cannot quit until all interviews have been completed or terminated and all stations are signed off (status not RUNNING, MONITOR, ACTIVE or ENDEDINT on DAI screen) unless they use the EMERGENCY_QUITNOW command.

EMERGENCY_QUITNOW Quit out of Supervisor

This is used to quit out of the Supervisor even if interviewers are logged on. If you do use this command, however, the socket used by the Supervisor will not be closed properly. So, you should use it to restart using a new LDEV value, such as "super 23" instead of "super 22.

SERVER:DOWN Closes all studies and takes the CfMC server down

This will only take the server down if there are no live interviewers. Use one of the interviewer stop commands to stop interviewers first or see SERVER:DOWNNOW below.

SERVER:DOWNNOW Closes all studies and shuts down the server

Use this with caution, as it stops the server regardless of whether it is busy doing something else or there are live interviewers. Also closes all connections to dialers and Sound Survent.

INTERVIEWER CONTROL

START_STATIONS (START) Start station(s) on a study

START (STS) authorizes stations for interviewing on a particular study. After an interviewing station has been authorized, interviewing may begin with the use of Survent. The interviewer typically logs onto the system and is automatically started under Survent, or enters START at the command prompt to set the CFMC variables and run Survent.

Here is the syntax and an example for authorizing interviewing stations to work on a particular study:

```
STS #ldevs <study> <mode>
EX:
STS 14-22 JV7922
```

Mode may be: CONT, DIALER/-DIALER (or ATD/-ATD), PRACTICE (or P or T), ROTATE_PRACTICE/-ROTATE_PRACTICE, SPECIAL=#, VALIDATOR (or V), or WAIT.

These can be specified in any order. A comma is required after the station list if a mode is specified. Here is more information on the parameters:

CONT Causes an interviewer to skip the between interviews prompt when interviewing. The interviewer would need to be killed or cleared to be unassigned from the study.

DIALER DIALER/-DIALER (or ATD/-ATD) assigns the station to use or not use the autodialer (EIS or MSG PRO-T-S).

PRACTICE (P or T for Training). This runs the interview in training mode. No data is saved. If Practice mode is specified, you can also use the keyword “ROTATE_PRACTICE” (RP) or –RP to control whether rotations will rotate while in practice mode.

SPECIAL= Designates the interviewer as one or more of the nine special interviewer types or none (0). This will override whatever is specified for the interviewer in the Employee Information file or what the interviewer might specify on the interviewer ID prompt in Survent.

VALIDATOR Start the interviewer as a validator. This is used to transfer records that have been completed by an interviewer to a validator to review the name and address, etc.

WAIT This is used when connecting with interviewers that have started themselves up in “WAIT” mode using NETSURV. The program assigns the study to the waiting interviewers so they can enter their interviewer ID and begin. WAIT mode is used in cases where there is no pre-assigned device number so you would not otherwise know which station to start.

```
EX:
STS 100, exam2,special=129
```

This example will override whatever is in the employee DB file with types 1, 2, and 9.

Commas are not allowed between the special interviewer types.

```
EX:
```

```
STS 100, exam2,special=129,practice,-atd
```

This example will set the special interviewer type to 1, 2, and 9, turn on training mode, and turn the autodialer off.

NOTE: You can also replace STS with ATM (or ATMS) to start up interviewers with the autodialer.

```
EX:
ATM 50-54 bank
```

SUPER can attach devices that have been started using NETSURV (shared files mode); this is done by having the interviewer press **Enter** at the "QFF file name-->" prompt nstead of filling in the study name. The device is listed as "waiting" in the DAI listing, and the supervisor can start them on a particular study. This is designed for systems where interviewers sign on using TELNET or some other system whereby the device number is initially unassigned; it allows the supervisor to find the device number once the interviewer is logged in and in Wait mode.

CHANGE_INTERVIEWER (CHI) Change interviewer(s) to a new study

This has the functionality of the combination of the STP (stop an interviewer on the current study at the end of the current interview) followed by a new STS (start an interviewer).

If you use CHI on an inactive interviewer, it acts like STS to start them. The syntax for changing the interviewer to another study is:

```
CHI <ids|#ldevs> <study>
```

You have the same options for <ids|#ldevs> and <study> as you do on the STS command.

The Supervisor can also use CHI to change interviewers to a new version of a questionnaire. They would specify the name of the new QFF file to start, but the file has the same internal study code as the existing study; this allows you to switch interviewers to a new version of the questionnaire while the old version is still active.

In addition, you can control whether interviewers are running with or without "special interviewer type" mode when using the "CHI" command to change them to a new study.

The syntax is:

```
CHI <interviewer list> <studyname> <S=# or S=none> <(-)Dialer> <(-)Practice>
```

“Interviewer list" can be a list of name or range of numbers, "studyname" can be any 3 to 30 character study name. "S=#" can be "-1" or "none" to turn off special type, or as many of the numbers 1-9 as special types assigned, "Dialer" or "-Dialer" are on/off switches, indicating whether or not a dialer is being used, and "Practice" or "-Practice" are on/off switches indicating whether you are or are not in practice mode.

The "CHI" command defaults to "LIVE" mode in all cases. If you want to change studies by saying "CHI" the program will put you in "Live" mode unless you say otherwise. Previously, if you were in "Practice" mode you stayed in practice mode when you said "CHI" unless you also entered "-PRACTICE".

GATHER Connect unsupervised interviewers to this supervisor

This is useful to attach unattached interviewers to this supervisor. One example is if the supervisor program has crashed for some reason and you need to reattach the interviewers that were running under the supervisor so you can start and stop them, see them in the info screens, etc. The syntax is:

```
GATHER <#ldev or Interviewer Ids>
```

BREAK Put interviewer(s) into Break mode

This happens as soon as the current interview is completed. If the interviewer is at the in-between interview prompt, they will go to Break mode immediately. This puts “Interviewer on Break” on the interviewer screen, “On Break” on the supervisor screen, and sends a log record to record the time interviewers went on break. Times can be be displayed using the log reports. The syntax for putting interviewers into Break mode is:

```
BREAK <ids|#ldevs|ALL/EVERYONE <study>>
```

If you say “BREAK EVERYONE Mystudy” everyone on study Mystudy will be prompted to go on break after they complete their current interview across all supervisors. If you say “BREAK ALL Mystudy” only interviewers under your control will be prompted to go on break for that study.

LUNCH Put interviewer(s) into Lunch mode

This occurs when the current interview is completed. If the interviewer is at the in-between interview prompt, they will go to Lunch mode immediately. This will put the message “Interviewer at Lunch” on the interviewer screen, “Out to Lunch” on the Supervisor DAI screen, and sends a log record to record the time interviewers went to lunch. Times can be displayed using the log reports. The syntax for putting interviewers into Lunch mode is:

```
LUNCH <ids|#ldevs|ALL/EVERYONE study>
```

If you specify “LUNCH EVERYONE Mystudy” then everyone running study Mystudy under the server will be notified to go to lunch. If you say “LUNCH ALL Mystudy” all interviewers started by you will be notified, but not other supervisors’ interviewers.

STOP Stop station(s) at end of interview

Allows you to stop an interviewer session after the current interview is completed. A message will appear on the interviewer’s screen at the end of the interview, and interviewing will terminate at that station. If the interviewer is currently in-between interviews, they will be stopped immediately. This command cannot be overridden once issued. Here is the syntax and an example for stopping an interviewer session:

```
STP <#station, interviewer ID, study name, ALL,
or EVERYONE>

EX:
STP #27
```

Indicate which interviewer to stop, either by entering the station number, interviewer ID, the study name, ALL to stop all interviewers under this supervisor, or EVERYONE to stop all interviewers under all supervisors. You must specify a study name when using EVERYONE.

```
EX:
STP EVERYONE MELMAC
```

STOP_NOW Forces an interview to "BLOW" and terminate.

This command does the same thing as the "KILL" command, except it saves the data up to this point in the ".B_" directory and saves a suspend record if the "AFTER_BLOW" command says to "SUSPEND". You can use "STOP_NOW #-#,#,#,#-#" etc. using the usual stations list.

KILL Kill station(s) after next interviewer keystroke

Sends a message to an interviewer’s screen and terminates the current interview after the next interviewer keystroke. Here is the syntax and an example for killing a station immediately:

```
KILL <#station-station, interviewer ID,
ALL/EVERYONE studyname >
```

#-# is a list of device numbers.

Interviewer ID can be a list of interviewer IDs.

ALL studyname means all interviewers under this supervisor for that study.

EVERYONE studyname consists of all interviewers under all supervisors for that studyname.

```
EX:
KIL #27
KIL EVERYONE MELMAC
```

When the interviewer presses **Enter** for the next question, the termination message will appear on the screen and they will be stopped immediately.

```
Session stopped by Supervisor at ldev #23
```

Note that the current interview will *not* be saved to the data file and the phone record will *not* be returned to the phone file with a status — it will be up-in-the-air. Use this command only at the end of the shift or as an emergency measure. STP should be used for stopping an interviewer, saving the current interview and returning the phone record to the phone file with an appropriate status. This command cannot be overridden once issued.

CLEAR Clear station(s) immediately

Regardless of whether active or interviewing, CLEAR removes the station from the server. It causes the entry in the “stations” file (\CFMC\IPCFILES\STATxxxx.IPCFILES) for the station(s) to be removed. Here is the syntax and an example for clearing a station:

```
CLEAR <ids|#ldevs>

EX:
CLEAR #22
```

You can only clear stations that are signed on under this supervisor. If you want to clear other devices (monitors, supervisors, or other supervisors’ interviewers), use the SERVER:CLEAR command described below.

SERVER:CLEAR Clear any station(s) from the server

Clears station(s) entry from the server stations file. This should only be used after attempts to STOP or KILL the station have been unsuccessful. You can use this command for other supervisor’s interviewers. The syntax for SERVER:CLEAR is:

SERVER:CLEAR <#ldevs>

SERVER:CLEAR_EXTENSION ## command clears stations by their phone extension instead of device number

You can clear a session by its phone extension instead of the device number. This is particularly useful for webCATI as the device numbers are assigned at startup. But the phone extension is pre-assigned and known for each interviewer.

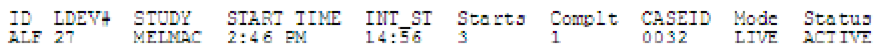
INTERVIEWER RELATED

DAI Display Active Interviewers

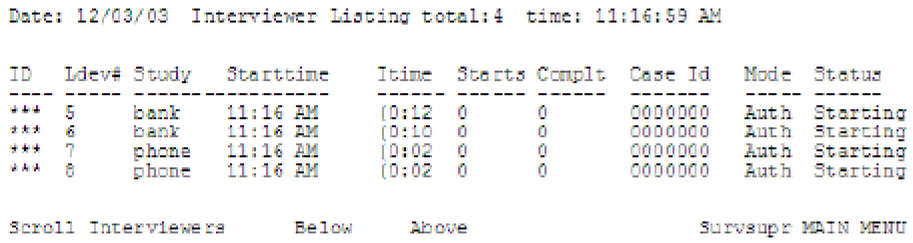
Lists station information about each interviewer known to SURVSUPR. The syntax for displaying active interviewers is:

```
DAI
```

By default, the following information is provided:



```
ID  LDEV#  STUDY   START TIME   INT_ST  Starts  Complt  CASEID  Mode  Status
ALF 27     MELMAC  2:46 PM      14:56   3       1       0032    LIVE  ACTIVE
```



```
Date: 12/03/03  Interviewer Listing total:4  time: 11:16:59 AM

ID   Ldev# Study    Starttime    Itime  Starts Complt Case Id   Mode  Status
---- ----- -----------------     ------ ------ ------ -------   ----- ------
***  5     bank     11:16 AM     00:12  0      0      0000000   Auth  Starting
***  6     bank     11:16 AM     00:10  0      0      0000000   Auth  Starting
***  7     phone    11:16 AM     00:02  0      0      0000000   Auth  Starting
***  8     phone    11:16 AM     00:02  0      0      0000000   Auth  Starting

Scroll Interviewers      Below     Above                 Survsupr MAIN MENU
```

where:

ID *** is the interviewer ID.

LDEV# 5, 6, 7, 8, are the ports or station numbers; on a DOS network, this is the node number; on a UNIX system, this is the station's corresponding number in the /CFMC/CONTROL/TTYINFO file.

STUDY bank and phone are the interviewer's assigned questionnaire study names.

STARTTIME 11:16 pm is the session start time.

ITIME Time of last interview start.

STARTS Number of interview starts this session

Complt Number of cases completed this SURVSUPR session.

CASE ID 0032 is the ID of the last case written to the data file by this interviewer. If the interviewer is in Training mode, TRAINING will show in this field

Mode is one of the following:

- Auth Authorized to start, but not started yet
- Train Live interview, but in training mode

- Live Currently doing live interview

Status ACTIVE is the station's status.

Here are the valid statuses:

Status	**Description**
Doing Interview	Interviewing
MONITORED	Interviewing and being monitored
Between	Interviewer is at the end of interview prompt
Quitting	Interviewer quit normally
BLOWN	Interviewing has been stopped due to a blow case
Before first interview	SERVER has acknowledged the existence of the station after it has been authorized by SURVSUPR
Out to Lunch	Supervisor or Interviewer has used the LUNCH command
Back from Lunch	Interviewer has resumed interviewing after LUNCH
On Break	Supervisor or Interviewer has used the BREAK command
Back from Break	Interviewer has resumed interviewing after BREAK

In addition, the first 20 characters of the last SPC,E,1 question seen in an active questionnaire will be shown in this field. This allows you to track the interviewer's progress within the questionnaire.

There is an optional parameter that you can add to get additional information. You add the type of interviewer you want listed and the study.

```
DAI <type> <studyname>
```

Type	**Description**

ALIVE	Lists all active interviewing stations
DEAD	Lists interviewers who have Quit or been killed
ALL	Lists all interviewers on the system, even ones not your own (i.e., controlled by other supervisors)
MINE	Lists all interviewers under this supervisor for a particular study.

Specifying LP or LP! at the end of the command will send a copy of the screen to the file indicated on a previous LPDEV command. The default DAI screen is updated every 15 seconds with new interviewer information as long as DASHBOARD:NO is not set in your TTYINFO/PARMFILE. Use the SURVSUPR command >DUMP a6 to disable this updating, and >DUMP a7 to enable it again.

If you use the ALIVE option, you'll get the standard DAI screen with this additional information of End time, Mode, S, and Ltime.

End time End of last interview, or end of session if Quit

Mode (see below)

S Special interviewer type

Ltime Time since last interview start

Mode+	**Description**
Dialwait	Waiting for EIS predictive dialer to return phone number
Live	Currently doing live interview
Active	Active station but not interviewing
Ended	Interviewer is at end of interview prompt
Killed	Interviewer has been killed by SURVSUPR
Quit	Interviewer quit normally
Blown	Interviewing has been stopped due to a blow case
Started	SERVER has acknowledged the existence of the station

Starting	SURVSUPR has acknowledged the existence of the station after it
No study	SERVER was never able to load study requested for that station
Stopped	Interviewer has been stopped by SURVSUPR
ID wait	At "enter interviewer ID" prompt

There is a subset of options to the DAI command when displaying station information, as well.

DAI SUPER 88 Displays all interviewers owned by supervisor 88

DAI 1-20 Display interviewers at stations 1 - 20

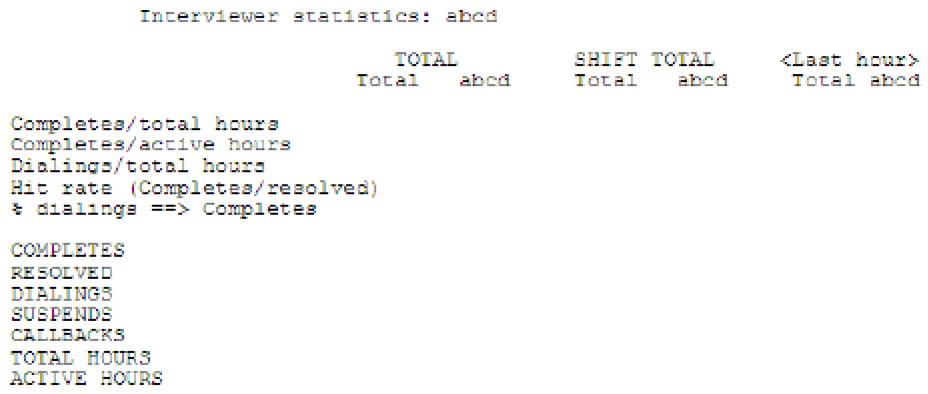
DAI 1 Display the interviewer at station 1

Optionally, to display more information on a particular interviewer, use:

```
      DAI  STATS <intv ID>
```

NOTE: The "Dashboard" parameter must be set in the study header of any study you want to get interviewer statistics on. Note that this requires additional overhead for the CfMC server.

This shows you the following:



```
          Interviewer statistics: abcd

                                      TOTAL           SHIFT  TOTAL       <Last hour>
                                   Total    abcd      Total    abcd       Total  abcd

Completes/total hours
Completes/active hours
Dialings/total hours
Hit rate (Completes/resolved)
% dialings ==> Completes

COMPLETES
RESOLVED
DIALINGS
SUSPENDS
CALLBACKS
TOTAL HOURS
ACTIVE HOURS
```

NOTE: The shift totals are reset to zero by using the END_SHIFT command.

DAITS shows the process ID and other information for terminals such as DAIWS and DAIWC FOR webSurvent and webCATI stations. This can be useful, for instance, to be able to figure out which process to kill in case there is a software abort and the station is hung up.

DAI ALL Display (all) Active Interviewers

Lists interviewers (one per line), and adds the current status of the call (eg. between interviews) and the interviewer special type (1-9). You can also see when the last interview started or ended for each interviewer. And, when the interviewer quits, it notes the time they stopped interviewing and the time they started up again, or a new supervisor session starts.

DAIWS Display Active Interviewers for webSurvent

It lists the LDEV, which is the CfMC assigned device number, the Study (study name), Password (the password on that sample record, PID (the Survent process ID for that session), Last (the last time the session contacted the server) and Qffname (the questionnaire file name). If there are no websurvent sessions, it will say so.

The format looks like this:

```
LDEV    Study        password      PID    last    qffname
10004:   rvsurvey    NOMCBI        20087 10:54:52 rvsurvey.qff
```

If you say "Daiws " it will just show the stations for a particular study.

DAIWC Display Active Interviewers for webCATI

It functions the same as DAIWS above. The format is also the same as DAIWS.

DATEFORMAT Controls date displays. When you use DATE_NO_LEADING_ZEROS with this option, you can control whether you see “5/5/07” or “05/05/07” in displayed dates.The default is to show the leading zeroes.

```
DATEFORMAT: DDMMYY DATENOLEADINGZEROS
```

This will show a European date display (DDMMYY) without the zeroes, such as "7/3/54" for March 3, 1954.

MONITOR (MON) Monitor a station

Allows any interview in progress to be monitored from any supervisor or external monitor using SURVMON (see *5.8 MONITORING WITH SURVMON)*.

If no station or interviewer ID is entered after the MON command this command will list the active interviewers (by station number and ID) and their current status (in interview, between interviews, etc.) If you have SPC,E,1 statements in the questionnaire, their text will be displayed to show the monitor at which question they are. In this way, the supervisor can see the progress of each active interviewer to help choose which one to monitor.

Here is the syntax and an example for monitoring a station:

```
MON <#station or interviewer ID>

EX:
MON #25
```

An interviewer may be monitored simultaneously by any number of monitors. See the SURVMON command for information on how to control whether a monitor can monitor other stations or studies.

When monitoring, the interviewer screens are displayed along with their answers. The word MON and the interviewer’s ID displays in the lower right corner of the monitor’s screen during monitoring. The compiler command -ALLOW_MONITOR disallows monitoring for the following questions, the monitor screen will be blank until moving beyond those questions.

Text question responses (long open-ends) are updated every 20 characters typed. You can control the length of time the answer remains on the screen with the Wait command (see below).

If you are monitoring and using a dialer, and the monitoring station has a phone extension on the dialer, the program will also play the monitored call's voice on the phone.

End the monitoring session by pressing **Ctrl-BREAK** (UNIX) or **Ctrl-Y** (DOS).

PRACTICE_MODE disallows users from aborting, or using "goto" if they are in practice mode

This keyword allows -GOTO, -ABORT and -LLRECS to control what happens when practice mode interviewers are being used. It also controls whether or not to make log records for the practice mode interviews.

TELL Sends message to station(s)

This allows the supervisor to send messages to station(s). The message will appear at the bottom of the interviewer's screen when the next **Enter** or **ESC** is pressed, or at the next screen update if the interviewer is between surveys. The message can be up to 77 characters. It will disappear from the screen when a subsequent **Enter** or **ESC** is pressed at the interviewer's machine. Here is the syntax and an example for sending a message to a station:

```
TELL #stations|intv Ids|<ALL/EVERYONE studyname>
<message>

EX:
TELL #15 NEW YORK quota has been changed to 200.
```

This would cause the following to appear on station 15's screen:

```
From Supervisor: NEW YORK quota has been changed to
200.
```

ALL is all interviewers on a particular study under this supervisor. You must specify a study name when using ALL.

EVERYONE indicates all interviewers under all supervisors. You must specify a study name when using EVERYONE.

```
EX:
TELL EVERYONE BANK Shift is over. Go home!
```

To display a message at the end of the current interview, not during the interview, put an E and a space before the message.

```
EX:
E Good job!
```

QUOTA RELATED

QSS Show quota screens or modify quotas for study

Allows you to show or modify the named quotas for any of your studies. Here is the syntax and an example for *showing* quotas:

```
QSS <study> <keyword> <pattern>

EX:
QSS bank SORT
```

Keywords are:

- **MODIFY** lets you make changes to the quotas unless the study header option MODIFY_QUOTA_FILE=N has been specified. If MODIFY_QUOTA_FILE=P is set and you try to modify quotas, you will be prompted for the password (as specified on the PASSWORD option on the header statement). Use QUOTAMOD to modify quotas if the Supervisor cannot modify quotas. You will be asked if you want to save changes after making any.
- **SORT** sorts the display in ASCII order
- **NOSORT** shows quotas in order specified in questionnaire. This is the default.
- **SHOWZEROVALUES** shows all quotas, even those with zero values. This is the default.
- **SHOW** shows just non-zero quotas.
- **RESET** allows you to reset the "resettable" quotas to zero for a study with a single command. Previously, you had to fill in zeros on each quota. The resettable quotas increase along with the regular quotas, but they are designed to allow you to reset them to zero at the beginning of some period so that you can

keep track of that period, separate from the overall quotas. The command is QSS <STUDY> RESET. This matches the "RESET" command in the quotamod program.

Furthermore, after specifying MODIFY, the supervisor moves the highlighted box (using arrow keys or **Ctrl**-**U**, **D**, **R**, and **L** as explained under MPF) to the desired quota and enters the new value. To move to the next item, you can hit ENTER.

If there are more quotas than will fit on one screen, use the **Page Up** and **Page Down** keys on monitors, and **Ctrl**-**T** (previous screen) , or **Ctrl**-**Y** (next screen) on terminals. You will see a message at the bottom of the screen indicating the number of pages of information.

Also, for UNIX, you can use **Ctrl**-**G** (top of list) **Ctrl**-**E** (end of list) or **TAB** (down). For DOS monitors, use **HOME** (top of list), **END** (end of list) and **TAB** (down).

NOTE: The old keystrokes **Ctrl**-**F**, **U** (up) or **Ctrl**-**F**, **D** (down) on UNIX and **ENTER** (next item), **right arrow** (move right), **left arrow** (move left) and **up arrow** (move up) in DOS still work. Some of the keystrokes were added to make it easier to get from screen to screen.

<pattern> allows you to specify a subset of quotas to show:

* Says any matching characters in name

? Says any matching alphabetic character in name

# Says any matching number in name

```
EX:
CITY@  would show all quotas that start with the word CITY
AREA.#  would show quotas AREA.1, AREA.2, etc.
ZONE.?  would show ZONE.A, ZONE.B, etc.
```

(Also refer to *5.5: QUOTAMOD*.)

After displaying the quotas on the screen, you can hit “Esc,” and the program will display a summary of time counts spent on the study from the quota file. Your display will look like this:

```
*survent_totaltime              = 000123:28:36
*survent_currentime             = 000017:41:14
*interviewing_totaltime         = 000102:14:05
*interviewing_currentime        = 000012:38:57
*unique_number                  = 0
```

This shows that there has been 123 total hours of interviewing on the study, 17 hours in the current session, 102 total hours in interviews, and 12 hours in interviews in the current session. The “unique_number” is a special quota value you can set in the program using !SPC,A,N to allow you to have a new number in each data record. It will update by 1 every time it is used. (See *!SPC,A, in Survent Chapter 3* for more information.)

Specifying LP or LP! at the end of the command will send a copy of the screen to the file indicated on a previous LPDEV command.

The default QSS screen is updated every 15 seconds with new interviewer information.

Quota changes are logged in the Server log files.

Modify Quota (MQ)

Similar to NQ below, modify_quota can be used to modify quotas, one at a time, in batch mode.

```
MQ <study name><quota number or name><+ - =><value>

      EX:
      MQ J4321 100=250
```

You can modify a set of quotas by using a quotaname pattern on the modify command:

```
EX:
MQ mike *.T =200
```

This would set the target values for all quotas to 200. Notice that you can use the "*" (all characters), "#" (single number), and "?" (single letter or number) as pattern matching characters.

NUMBERED_QUOTA_SHOW (NQS) Show/Modify Numbered Quotas on screen

Allows you to show and/or modify numbered quotas. Specify the study name and quota number to start at. The screen displays quotas, beginning with the numbered quota you specified and continuing for up to 250 quota cells. If you say MOD, modification of quota values is allowed. If you say ZERO then quota cells with a zero (0) value are shown, otherwise they are not shown.

Here is the syntax and an example for changing the numbered quota show:

```
NQS <study name> <number> MOD | ZERO

EX:
NQS J4321 725 MOD
```

NUMBERED_QUOTA (NQ) Modify Numbered Quotas by command

Allows specific numbered and named quota modification in batch mode at the command line, without requiring you to enter the quota screen for modification. Here is the syntax and an example for changing the numbered and named quotas:

```
NQ <study name><number or name><+ - =><value>

EX:
NQ J4321 100=250
```

SUPERVISOR/FONEUTIL-RELATED

SURVSUPR now supports most FONEUTIL commands while live. Most of the commands that previously could only be run offline while a study is down may now be done in the supervisor while the study is live. Some of these commands also are incorporated into the webSuper Utilities program. (See the *webSuper Manual* for more on these commands).

The commands supported are:

PHONE_LIST <study><select>: List sample to screen or to LPDEV if specified.

PHONE_ZAP <study> <Last Save> <select>: Remove call histories and make sample "new". "Last" removes last call history only. "Save" removes no histories but changes stack.

PHONE_ERASE <study> <select>: Same as Zap but only on active sample.

PHONE_KILL <study> <select>: Resolve sample with status 81 immediately.

PHONE_ALTER_TIMED <study> <select> <time>: Changes time of callbacks.

PHONE_RETURN_OWNED_NUMBERS <study> <select>: Returns numbers from "owned" stack to original stack.

PHONE_GATHER_SPECIALS <study> <select>:Move numbers from other stacks to "special interviewer" stacks.

PHONE_SORT_SPECIALS <study> <1-9>: Sort "special interviewer" stacks by time zone for proper call order. Use "1" for instance to sort special interviewer type 1 stack **only.**

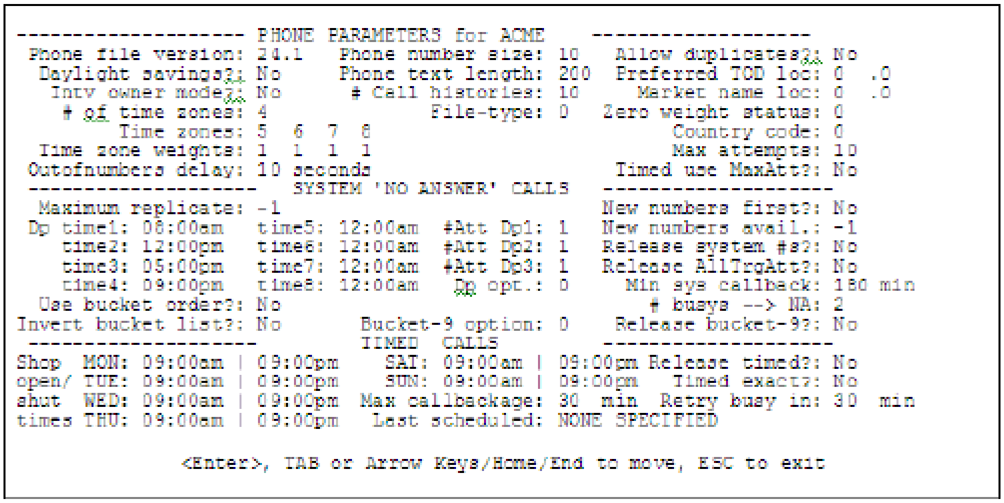
SAMPLE-RELATED

MODIFY_PHONE_FILE (MPF) Modify phone parameters for study With this option you may modify most phone file control parameters. Here is the syntax and an example for modifying a phone file:

```
MPF <study>

EX:
MPF JOB123
```

Position the cursor on the item you want to modify and then enter in the new value. Use arrow keys to position the cursor on monitors and **Ctrl-U** (up) and **Ctrl-D** (down; **R** and **L** will move right and left) on terminals to move the highlighted box to the appropriate item on terminals. Press **ESC** and you will be prompted as to whether to save the changes, then you will return to run mode.

MODIFY_FONE_FILE=Y/N/P in the questionnaire header controls whether or not supervisors can use the MPF option. If allowed to use it, they must know the password assigned on the PASSWORD option on the study header. If not allowed to modify the phone file, supervisors can use the SPI command to look at the current setting. You can use FONEUTIL to make phone parameter modifications in any case while the study is inactive.


```
-------------------- PHONE PARAMETERS for ACME    -------------------
 Phone file version: 24.1   Phone number size: 10   Allow duplicates?: No
  Daylight savings?: No     Phone text length: 200  Preferred TOD loc: 0  .0
   Intv owner mode?: No      # Call histories: 10     Market name loc: 0  .0
    # of time zones: 4              File-type: 0  Zero weight status: 0
         Time zones: 5  6  7  8                         Country code: 0
  Time zone weights: 1  1  1  1                         Max attempts: 10
 Outofnumbers delay: 10 seconds                     Timed use MaxAtt?: No
 --------------------   SYSTEM 'NO ANSWER' CALLS   --------------------
  Maximum replicate: -1                            New numbers first?: No
 Dp time1: 08:00am   time5: 12:00am  #Att Dp1: 1   New numbers avail.: -1
    time2: 12:00pm   time6: 12:00am  #Att Dp2: 1   Release system #s?: No
    time3: 05:00pm   time7: 12:00am  #Att Dp3: 1   Release AllTrgAtt?: No
    time4: 09:00pm   time8: 12:00am   Dp opt.: 0     Min sys callback: 180 min
  Use bucket order?: No                                 # busys --> NA: 2
Invert bucket list?: No       Bucket-9 option: 0    Release bucket-9?: No
 --------------------         TIMED  CALLS          --------------------
Shop  MON: 09:00am | 09:00pm    SAT: 09:00am | 09:00pm Release timed?: No
open/ TUE: 09:00am | 09:00pm    SUN: 09:00am | 09:00pm   Timed exact?: No
shut  WED: 09:00am | 09:00pm  Max callbackage: 30  min  Retry busy in: 30  min
times THU: 09:00am | 09:00pm   Last scheduled: NONE SPECIFIED

             <Enter>, TAB or Arrow Keys/Home/End to move, ESC to exit
```

Items with "*" to the left of them may not be modified after the original phone file has been constructed. For an explanation of each of the elements above, see *6.1.1 BUILDING THE SHOP AND PHONE FILES.* The phone file is immediately updated after every MPF command if UPDATEFONEHEAD:Yes is specified in the TTYINFO/PARMFILE; otherwise, some commands may not take effect until interviewers are restarted. When you exit the MPF screen, you will be reminded of the changes just made, and asked whether to save the changes. Changes are logged to the SERVER log file.

NOTE: By making inappropriate changes on this screen, you can stop phone numbers from being available to the interviewers. Make sure you are changing the right parameter in the right way.

DISALLOW MPF lets you control what can be changed on the MODIFY_PHONE_FILE screen.

The "disallow mpf xxxx" command can be placed in the support/suprinit file and those command will be disallowed.

The syntax is:

```
DISALLOW MPF <field name>
```

The field names must match the name you see when the program prompts you with the "are you sure?" prompt. Here are the field names that you can use and their meanings:

Field name	**Description**
minage	minimum age of system calls
busyresked	busy reschedule time
numfromnever	numbers from fresh buckets
maxattempts	maximum attempts
numbusyna	Number of busy numbers at which return no answer
ownermode	owner mode
time1bucket	daypart 1 start time
time#bucket	daypart 2 - 8 start times
openshut	open and shut times (Monday-Sunday)
shutofftime	time when numbers will no longer be released
timed_limit	time after which numbers cannot be scheduled
freshfirst	whether fresh numbers are called first
invert_bucketlist	invert bucket list so call 7-1 instead of 1-7
callbackage	time after which uncalled timed numbers are put back

tzweight	time zone weights (1-8)
outofnumdelay	out-of-numbers delay time in seconds
release_timedcalls	whether to release timed calls now
exact_timedcalls	Get timed calls at exactly their time, not within 5 min
release_alltargeted	whether to released previously called numbers
release_allbuckets	whether to release all "no answer" and new numbers
release_bucket9	whether to release bucket 9 holding numbers
replicate	value to set as maximum replicate
target	controls target number of attempts for dayparts 1-3
bucket9_option	this controls the use of the bucket 9 holding stack
timeperiod_option	controls the days considered for day parts
hardmaxattempts	controls whether max attempts applies to timed calls
market_zero_status	status to give timed numbers coming from zero-weight market

These can be used in addition to the other "disallow" commands.

CHANGE_OWNER (CHO) Change owner of sample records to new owner Causes all records owned by the specified old owner which are not hidden or resolved to be changed so they are owned by the new owner. The syntax for changing owners is:

```
CHO <study> <old owner> <new owner>
```

HIDE Hide selected group of phone numbers

This will hide all numbers in the .fon (sample) file that match a SELECT statement. See *6.7.1 MANAGING THE PHONE FILE WITH FONEUTIL* for more information on HIDE (as shown in FONEUTIL) and the SELECT statement. The SELECT statement uses the phone file format as explained under PHONE,G in *6.3.1 THE PHONE STATEMENT, PHONE, letter Subtypes*. Large phone files may take some time to hide all the requested numbers. The supervisor cannot quit until the hiding is complete. A message will display on the supervisor's screen when the hide is done.

Several HIDE operations can take place at the same time. Numbers that have been hidden may later be REVEALed. NOTE: You cannot hide "ALL" records but you can reveal "ALL" records.

Use FONEUTIL to HIDE numbers when the study is not live (outside of SURVSUPR). Also, see SAVE_SELECT later in tSample Related Commands:his section under Unlisted Options to see how to save SELECT statements for easier recall in REVEAL operations.

NOTE: You will not be allowed to hide "ALL" numbers in the Supervisor. But, if you really want to hide all the numbers you can do so by using a location specifications such as "[1#0-9] which specifies that all records with a number in the first position of the file (which would be all numbers since that is where the phone number is) will be hidden.

HIDE_REVEAL_STATUS (HRSTATUS) Show status of active HIDE/REVEAL operations

REVEAL (REV) Reveal previously hidden numbers

This will reveal previously hidden (see HIDE above) phone numbers which match a SELECT statement. See *6.7.1 MANAGING THE PHONE FILE WITH FONEUTIL* for more information on REVEAL (as shown in FONEUTIL) and the statement. The SELECT statement uses the phone file format as explained under PHONE,G in *6.3.1 THE PHONE STATEMENT, PHONE, letter Subtypes*. The numbers are revealed as the SERVER is available; large phone files may take some time to search. You can reveal "ALL" numbers if you choose.

The supervisor cannot quit until the revealing is complete. Several REVEAL operations can take place at the same time. Use FONEUTIL to reveal outside SURVSUPR.

Also, see SAVE_SELECT later in this section under Unlisted Options to see how to save SELECT statements for easier recall in REVEAL operations.

Also see NAMED_HIDE/REVEAL commands below.

MARKET Display numbers in market or modify market weights

This displays how many numbers are in each system bucket and total within a particular market or displays the total numbers in each market and their market weight. You can modify the market weight to call numbers in the market more or less often. Market weights may be 0-9; if they are set to 0 for a market it is not called. To display all markets or modify market weights use the command:

```
MARKET <study>
```

To display the status of a particular market is:

```
MARKET <study> <market>
```

To see only the markets with a weight > 0, use the command:

```
MARKET <study> -ZERO
```

The syntax for setting the market weight for the specified market is:

```
MARKET <study> <market>= <weight>
```

To export markets, use the syntax:

```
MARKET <study> MARKETWEIGHTOUT <filename>
```

This will export the market weights for <study> to file <filename>. These can be edited and read back in using the &filename command. This command falls under control of the MODFONE=Yes/No/Pass parameter in the study header to force users to use a password to use the command. See *6.3.4 Market Areas* for more information.

To print the standard time zone grid for all markets to the listfile and/or the screen, use:

```
MARKET <study> ALLMARKETS
```

To print one line per market, listing its market number, name, current weight and number of people in the market's available call grid, use:

```
MARKET <study> PRINTWEIGHTS
```

MINAGE_LIMITS This option sets the minimum and maximum values that a user can specify on the MINIMUM SYSTEM CALLBACK TIME parameter on the MPF screen. By default, the minimum is 0 and the maximum 9999.

For example:

```
      MINAGE_LIMITS: 0 180
```

This says that the minimum value users can specify is 0 and the maximum value is 180. If a supervisor sets the MIN SYS CALLBACK TIME to 0, numbers will be available immediately to be recalled. Note that, by default, you cannot set the numbers to be called back immediately. So, this is the only way to allow that.

NAMED_HIDE/REVEAL Allows naming the statement to reveal exactly those records, only later.

You can now use the "NAMED_HIDE" statement to hide numbers that you want to keep together as a group to be revealed later. This would be used, for example, to hide all the numbers in a quota group and then reveal them later by name. It makes it easier for staff to later reveal the group and it is independent of other hide/reveal operations. Note that these numbers have their own stack so that you can hide the numbers here, and they will be unaffected by the standard hide/reveal statements.

```
      Syntax:
NAMED_HIDE  <1-30 character name> [HIDDEN] <select statement>

      Examples:
Namedhide exam1 neworleans [51.2$]="NO"
Namedhide exam1 neworleans HIDDEN [51.2$]="NO”
Namedreveal exam1 neworleans ALL
```

Use the keyword "HIDDEN" To move numbers from the hidden stack as well, otherwise numbers in the hidden stack are ignored.

You can put a select statement on the "Named_reveal" to get a subset of the numbers returned. Notice that you can reveal multiple-named hides concurrently.

The "name" you specify is used when you later show a list of the current groups you are hiding so that you can reveal them. To display the list of available named hides, type "Namedhide <study> show".

PHONE_ADD_RAW_SAMPLE allows loading of raw sample like FONEBULD. You can now add new sample while a study is live from the supervisor using the "phone_add" command. The syntax is:

```
Syntax:
      PHONE_ADD_RAW_SAMPLE <study> <file to add>
```

The program will add all the valid numbers and print a summary of the records read, added and discarded. Numbers that are invalid are also displayed on the screen with an error code. The error codes are the same codes returned using the !PHONE,A statement to add sample (*see 6.3.1, The Phone Statement* for more information on !PHONE,A).

If you specified a "rejects_to_file" filename when the phone file was built in FONEBULD, any invalid records will be saved in that file as well for later processing.

PHONE_MOVE_RECORDS Allows you to move records from stack to stack using a "SELECT" statement to determine which records to move. Used in conjunction with the "Call_first" stack, this lets you give numbers immediate priority as well, if you choose. The syntax is:

```
         Syntax:

PhnMv Fromstack=<from stacks> Tostack=<to stack> <select
statement>
```

Stacks can be numbers or names. You can use up to 10 numbered stacks in the “from” field, but only one named stack.

Here are possible names:

From Names	To Names	Description
#	# (<350)	Stack number
CALLFIRST	CALLFIRST	“Call first” stack
FRESH	FRESH	
EMAIL_INVITE/REMIND	EMAIL_INVITE/REMIND	
TZ#B#		Time zone X, Bucket Y
TZ#B#M[#/marketname]		Time zone X, Bucket Y, market Z

Where "#" is a stack number. Stack numbers go from 1 - 9650. Notice that you can move things FROM most stacks, but you cannot move things to stacks > 350 (the time zone/market array). Users shouldn't move things from time zone to time zone or market to market in this way.

The "TZ#B#" syntax means "Time Zone X, bucket Y". The TZ#B#M# means "Time Zone X, bucket Y, Market Z". You can substitute a market name for the market number.

Also, there is a keyword "older_than" that will allow you to say "older_than 3 days" (to only get older records). Here are some examples:

```
phone_move_records fromstack=fresh tostack=callfirst [52.3$]="001"
phone_move_records fromstack=callfirst tostack=emailsend 1=1
phone_move_records fromstack=email_invite tostack=email_remind 1=1
phone_move_records fromstack=tz5b3mMymarket tostack=fresh 1=1
phone_move_records fromstack=tz5b3 tostack=fresh older_than 3 days
1=1

phone_move_records fromstack=fresh tostack=callfirst
                                          [1.10$]="4069954724"
phone_move_records fromstack=350 360 tostack=callfirst 1=1
```

SERVER_DO_NOT_CONTACT_FILE (SERVERDNC) identifies the file for the server to check new and changed phone numbers for callability.

The SURVSUPR command tells the server that for every phone number retrieved from any fonefile, look in the specified file and if the number is in there, put it back into the fonefile in the “nevercall” stack and go get another number. The server's DNC file can also be specified in the parmfile as "serverdnc: " The server's DNC file can be turned off or changed to another file at any time.

This is to allow you to do “real-time” checking of the phone file instead of just depending on what fonebuld did when it built the file looking at the standard DNC files.

STOP_HIDE_REVEAL (STPHR) Stop current HIDE, REVEAL, or CHANGEOWNER

Remember to use the HRSTATUS command first to see what the current operation is if more than one is running, or you may stop an operation you didn’t want to stop.

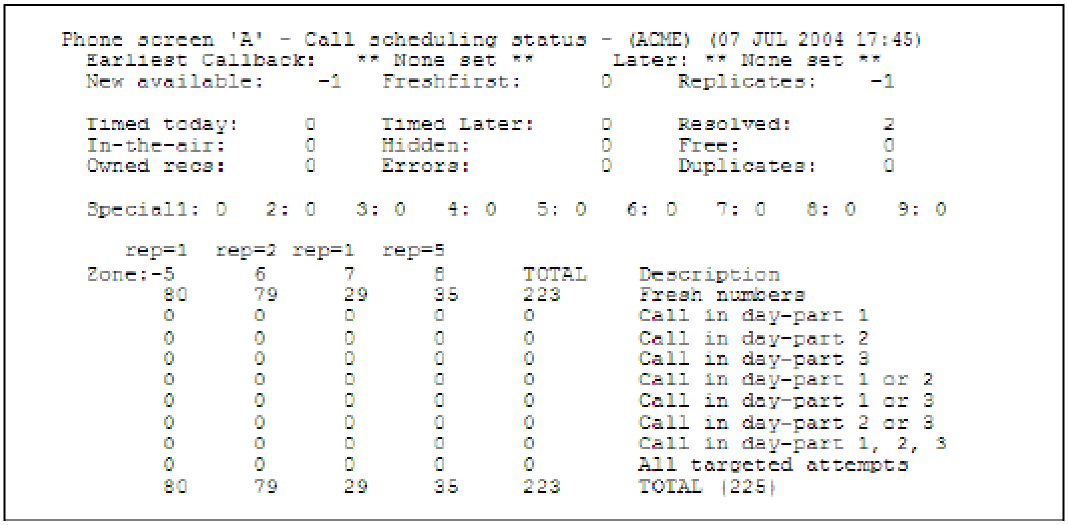
SAMPLE INFORMATIONAL

SPI Show Phone Information for study

This displays up to sixscreens of phone system information. Here is the syntax and an example for showing phone information:

```
SPI <a,b,c,d e or f> <study>

EX:
SPI JV7922
```

<a,b,c,d,e or f> is any of the six information screens available. The default, if no letter is specified, is all screens, one after the other, with an automatic timeout on each screen.



```
Phone screen 'A' - Call scheduling status - (ACME) (07 JUL 2004 17:45)
  Earliest Callback:    ** None set **       Later: ** None set **
  New available:     -1    Freshfirst:       0      Replicates:     -1

  Timed today:      0      Timed Later:      0      Resolved:        2
  In-the-air:       0      Hidden:           0      Free:            0
  Owned recs:       0      Errors:           0      Duplicates:      0

  Special1: 0   2: 0   3: 0   4: 0   5: 0   6: 0   7: 0   8: 0   9: 0

     rep=1  rep=2 rep=1  rep=5
  Zone:-5     6      7      8       TOTAL    Description
        80     79     29     35     223      Fresh numbers
        0      0      0      0      0        Call in day-part 1
        0      0      0      0      0        Call in day-part 2
        0      0      0      0      0        Call in day-part 3
        0      0      0      0      0        Call in day-part 1 or 2
        0      0      0      0      0        Call in day-part 1 or 3
        0      0      0      0      0        Call in day-part 2 or 3
        0      0      0      0      0        Call in day-part 1, 2, 3
        0      0      0      0      0        All targeted attempts
        80     79     29     35     223      TOTAL (225)
```

Description of information provided:

Earliest Callback Date and time of the next scheduled callback today.

Earliest Call Later Date and time of the next scheduled callback after today.

New available The total number of phone records which have yet to be attempted.

Timed Today Total numbers which are scheduled to be called back today, the earliest of which is Earliest Callback.

Timed Later Total numbers which are scheduled to be called on a day later than today.

Resolved Total numbers which have been resolved. This would include completed interviews, nonworking numbers, terminations, refusals, maximum number of attempts, etc. Resolved phone numbers will not be used again.

In-the-air Total records which have not been returned to the phone file with a status.

This would include currently active records and any records which were not written back to the phone file with a status, probably due to non-normal termination of the interview process. Use FONEUTIL's option V and U to return these numbers to where they came from.

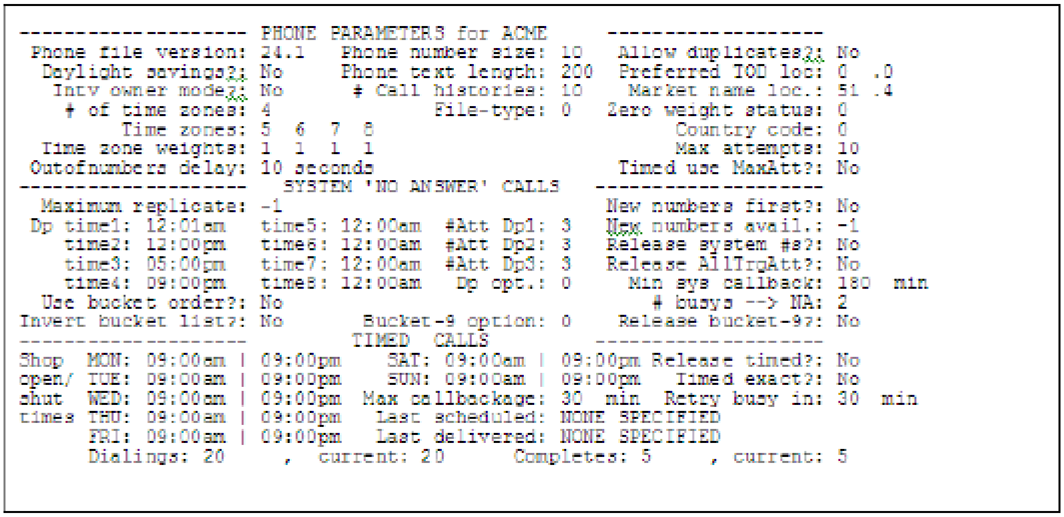
Hidden Total number of records which are presently *hidden*. Hidden records may not be dialed. (See FONEUTIL for more information on hidden phone records.) Use FONEUTIL's option R or SURVSUPR's REVEAL option to reveal these numbers.

Free Total number of available phone number slots in the phone file. Each number occupies a physical slot in the phone file. When a number is deleted using FONEUTIL, the slot for that number is then free. FONEBULD, when adding numbers to a phone file, first fills any free slots before appending to the file.

Special-1- 9 The total number of scheduled callbacks which will only be released to special type 1 (-9) interviewers. (See SPC,O and the employee information file for information on assigning special interviewer types.)

Rep=# Shows the replicate number of the first phone number in the fresh number stack for each time zone.

The table is a cross between time zones and categories for system scheduled numbers. The horizontal axis is divided into the different times zones for the study while the vertical axis is divided into the ten different categories. The values in the table represent the total numbers that satisfy both the time zone and category depending on the cell in the table. Each cell in the table also has a corresponding stack. (See *6.4 HOW THE PHONE SYSTEM SCHEDULES CALLS, System Scheduled Call Stacks*.)

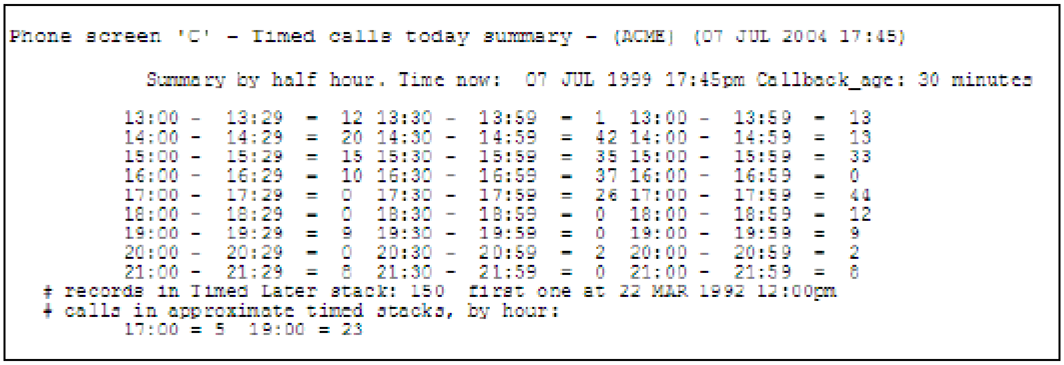


```
--------------------  PHONE PARAMETERS for ACME     -------------------
 Phone file version: 24.1   Phone number size: 10   Allow duplicates?: No
  Daylight savings?: No     Phone text length: 200  Preferred TOD loc: 0  .0
   Intv owner mode?: No      # Call histories: 10    Market name loc.: 51 .4
     # of time zones: 4              File-type: 0   Zero weight status: 0
         Time zones: 5  6  7  8                          Country code: 0
  Time zone weights: 1  1  1  1                          Max attempts: 10
 Outofnumbers delay: 10 seconds                      Timed use MaxAtt?: No
--------------------  SYSTEM 'NO ANSWER' CALLS   --------------------
  Maximum replicate: -1                             New numbers first?: No
 Dp time1: 12:01am  time5: 12:00am  #Att Dp1: 3   New numbers avail.: -1
    time2: 12:00pm  time6: 12:00am  #Att Dp2: 3   Release system #s?: No
    time3: 05:00pm  time7: 12:00am  #Att Dp3: 3   Release AllTrgAtt?: No
    time4: 09:00pm  time8: 12:00am   Dp opt.: 0     Min sys callback: 180  min
  Use bucket order?: No                                # busys --> NA: 2
Invert bucket list?: No       Bucket-9 option: 0    Release bucket-9?: No
--------------------         TIMED  CALLS         --------------------
Shop  MON: 09:00am | 09:00pm    SAT: 09:00am | 09:00pm Release timed?: No
open/ TUE: 09:00am | 09:00pm    SUN: 09:00am | 09:00pm   Timed exact?: No
shut  WED: 09:00am | 09:00pm  Max callbackage: 30  min  Retry busy in: 30  min
times THU: 09:00am | 09:00pm   Last scheduled: NONE SPECIFIED
      FRI: 09:00am | 09:00pm   Last delivered: NONE SPECIFIED
      Dialings: 20     ,  current: 20      Completes: 5     , current: 5
```

An explanation for each item above can be found in FONEBULD except for the following:

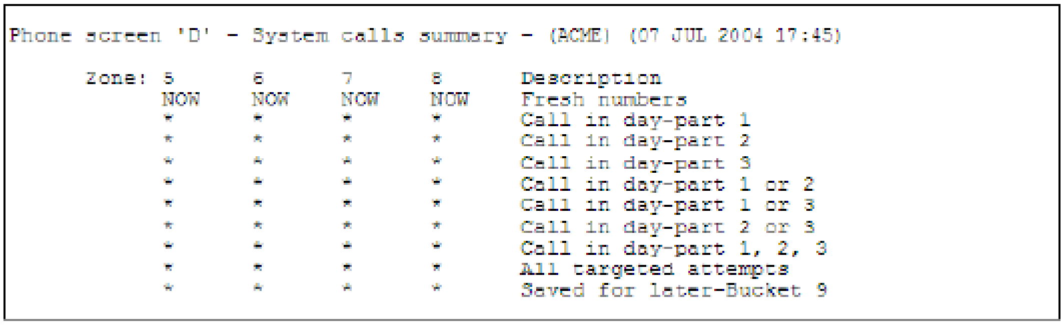
Dialings Total number of attempts made since the start of the study. (Current number of dialings is not implemented.)

Completes Total number of completes written to the data file since the start of the study. (Current number of completes is not implemented.)



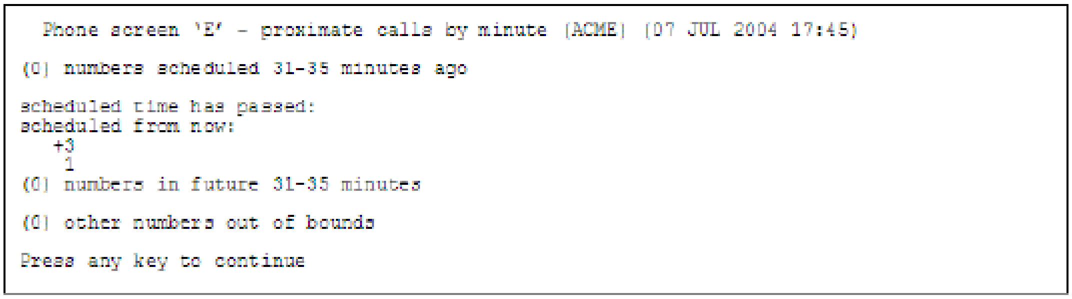
```
Phone screen 'C' - Timed calls today summary - (ACME) (07 JUL 2004 17:45)

            Summary by half hour. Time now:  07 JUL 1999 17:45pm Callback_age: 30 minutes

          13:00 -  13:29  =  12 13:30 -  13:59  =  1  13:00 -  13:59  =  13
          14:00 -  14:29  =  20 14:30 -  14:59  =  42 14:00 -  14:59  =  13
          15:00 -  15:29  =  15 15:30 -  15:59  =  35 15:00 -  15:59  =  33
          16:00 -  16:29  =  10 16:30 -  16:59  =  37 16:00 -  16:59  =  0
          17:00 -  17:29  =  0  17:30 -  17:59  =  26 17:00 -  17:59  =  44
          18:00 -  18:29  =  0  18:30 -  18:59  =  0  18:00 -  18:59  =  12
          19:00 -  19:29  =  9  19:30 -  19:59  =  0  19:00 -  19:59  =  9
          20:00 -  20:29  =  0  20:30 -  20:59  =  2  20:00 -  20:59  =  2
          21:00 -  21:29  =  8  21:30 -  21:59  =  0  21:00 -  21:59  =  8
   # records in Timed Later stack: 150  first one at 22 MAR 1992 12:00pm
   # calls in approximate timed stacks, by hour:
          17:00 = 5  19:00 = 23
```

This displays the timed numbers scheduled for today by half hour and hourly counts.



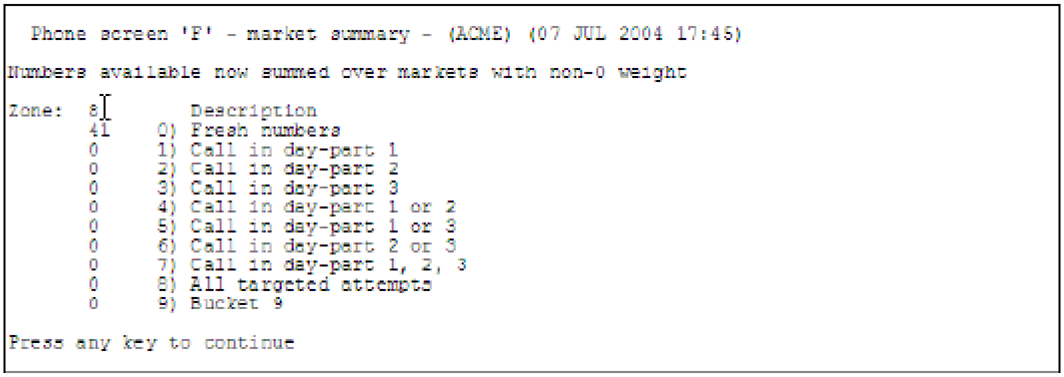
```
Phone screen 'D' - System calls summary - (ACME) (07 JUL 2004 17:45)

      Zone: 5       6       7       8       Description
            NOW     NOW     NOW     NOW     Fresh numbers
            *       *       *       *       Call in day-part 1
            *       *       *       *       Call in day-part 2
            *       *       *       *       Call in day-part 3
            *       *       *       *       Call in day-part 1 or 2
            *       *       *       *       Call in day-part 1 or 3
            *       *       *       *       Call in day-part 2 or 3
            *       *       *       *       Call in day-part 1, 2, 3
            *       *       *       *       All targeted attempts
            *       *       *       *       Saved for later-Bucket 9
```

This displays how soon calls will be available in each of the system callback stacks.



```
  Phone screen 'E' - proximate calls by minute (ACME) (07 JUL 2004 17:45)

(0) numbers scheduled 31-35 minutes ago

scheduled time has passed:
scheduled from now:
   +3
    1
(0) numbers in future 31-35 minutes

(0) other numbers out of bounds

Press any key to continue
```

This screen shows that there will be one call in three minutes from now and no numbers that have passed their time to call.



```
  Phone screen 'F' - market summary - (ACME) (07 JUL 2004 17:45)

Numbers available now summed over markets with non-0 weight

Zone:   8        Description
        41    0) Fresh numbers
        0     1) Call in day-part 1
        0     2) Call in day-part 2
        0     3) Call in day-part 3
        0     4) Call in day-part 1 or 2
        0     5) Call in day-part 1 or 3
        0     6) Call in day-part 2 or 3
        0     7) Call in day-part 1, 2, 3
        0     8) All targeted attempts
        0     9) Bucket 9

Press any key to continue
```

This displays the stack grid for the time zones used and tells you how many numbers are available across all markets. To see each market separately, see the MARKET command. See *6.1 INTRODUCTION TO THE PHONE SYSTEM* and following sections for information on the values and settings.

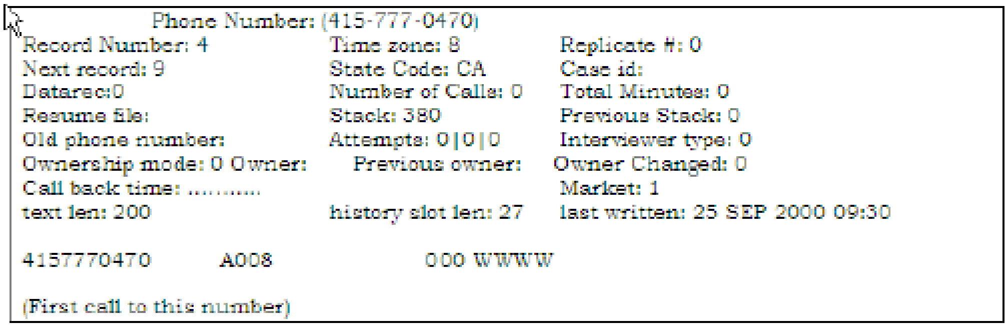
Specifying LP or LP! at the end of the command will send a copy of the screen to the file indicated on a previous LPDEV command.

The default SPI screen is updated every 15 seconds with new interviewer information as long as DASHBOARD:NO is not set in your TTYINFO/PARMFILE. Use the SURVSUPR command >DUMP a6 to disable this, and >DUMP a7 to enable it again.

PHONE_NUMBER (PHN) Displays information about a particular phone number in a study sample file. Here is the syntax and an example:

```
PHONE_NUMBER <study> <phone number or record number>

      EX:
      PHN acme 4157770470
```

Here is the information displayed:



```
                Phone Number: (415-777-0470)
Record Number: 4                    Time zone: 8              Replicate #: 0
Next record: 9                      State Code: CA            Case id:
Datarec:0                           Number of Calls: 0        Total Minutes: 0
Resume file:                        Stack: 380                Previous Stack: 0
Old phone number:                   Attempts: 0|0|0           Interviewer type: 0
Ownership mode: 0 Owner:              Previous owner:        Owner Changed: 0
Call back time: ..........                                    Market: 1
text len: 200                       history slot len: 27      last written: 25 SEP 2000 09:30

4157770470      A008              000 WWWW

(First call to this number)
```

LIST_PHONE_RECORDS (LIST) Print info on select group of phone numbers "LIST_PHONE_RECORDS" prints the same display as the foneutil "list" command, listing phone number information using the select statement to list the numbers you want to see. You can use "LIST" as shorthand. Here is an example:

```
LIST BANK
Enter SELECT statement or ALL--> [record_number] = 1
312-888-9988      ACTIVE: FRESH     ZONE: 06 STATE: IL
(First call to this number)
3128889988        06                Market 134 Male
Fred Jones        418 Small lane    Chicago, Ill 35462
```

SHOW_MANANA lists timed calls scheduled for the near future. With this command, you can now get a list of the timed calls scheduled for up to 60 days scheduled for the future. Here is an example:

```
Example:
      SHOW_MANANA PHONE

show manana_days 7
2 records at 09 MAR 2007 10:00
1 records at 09 MAR 2007 11:00
1 records at 09 MAR 2007 13:00
1 records at 10 MAR 2007 08:00
```

This is the only way to show calls scheduled beyond today. You may specify "manana_days=#" in FONEBULD when building the sample file to control how many days to display; where "#" can be a number from 1 to 60. The default is 7 days.

Survent can also get a report of these numbers that can be used to help with call scheduling applications. See the XF(TIMED_CALL) function in this manual for more information. Also, you may generate a report using PHONERPT to get these numbers offline or between shifts.

SHOW_PHONE_INFO F shows phone information on a subset of markets or a particular market. The new syntax is:

```
Syntax:
      SPI F <study> <market list>
```

The "market list" can be any standard alpha reference using wildcards, such as "me*" for all markets beginning with me, or just a list of the markets you want to see, as in "north,south".

This allows you to get summary numbers in the "F screen (how many numbers are available in stacks).

Note that the same subsetting feature can be used in the Foneutil "P F" option and the Fonesim "PZ F" option (these provide the same reports).

SHOW_STACK Shows the phone record information for the phone number at the top of a particular stack. The display is the same as that for the PHONE_NUMBER command.

Here is the syntax:

```
SHOW_STACK <study> <stack number or bucket,timezone>
```

STUDY CONTROL

SHUTDOWN_STUDY (SDS) Stop an inactive study

Drops a study from active memory. When interviewers quit working on a study, the study's files are still open by SERVER until the study is shut down by SURVSUPR (or the SERVER is taken down). Using the SDS command allows the supervisor to close all the files and start up another study. See DIS for summary of active studies. Here is the syntax and an example for shutting down a study:

```
SDS <study names or numbers/file pattern/ALL>
EX:
SDS JV7922
```

Studies can be removed by specifying the range of study numbers (i.e., 1-5) or "study name, study name" (i.e., exam2, bank), or by using the wildcard character "*" , or ALL to remove all studies.

```
EX:
SDS Bank*
```

This will shut down all studies beginning with Bank.

DOWN_STUDY (DOWN) Stop an active study and close study files

This is used to bring a study down even if the program thinks there are still interviewers working on it. Use this only after clearing stations and trying SDS (shut_down_study). This will close any of that study's files left open so they may be accessed by other programs. The syntax for DOWN_STUDY is:

```
DOWN_STUDY <study>
```

SERVER:CLEARSTUDY Clears all access to study files without having to bring the server down to get access to a study or file. The syntax for SERVER:CLEARSTUDY is:

```
SERVER:CLEARSTUDY <studyname>
```

LOAD_STUDY Loads a study under the server

It enables you to load studies under the server. Note that a studiy will also load automatically when you use any other study related command (eg. START, MPF, QSS, SPI). Use this command if you just want to load the study files so they are protected from overwrites and logged as being active.

```
    SYNTAX:
    LOAD <study name/name pattern>
```

If you specify a "name pattern", loadstudy will try to open all studies that meet that file pattern that it finds in the $CFMCQFL .qff file area.

The pattern can include "*" for matches to "all", "?" to match single characters, and "#" to match numbers.

```
    EX:
    LOAD phon*
```

This would load all studies that start with "phon" such as "phon1234", "phonabcd", and "phon_a12".

In addition, the LOAD command allows you to load multiple studies and file patterns by using a single LOAD command. The syntax is:

```
    Syntax:
LOAD <studyname/file pattern1,file pattern2,…,file pattern n>
```

There can be a single study, a file pattern or multiple file patterns. The program tries to load all the studies with that pattern with a .qff file in the $CFMCQFL directory.

STUDY INFORMATIONAL

DIS Display Information on Studies

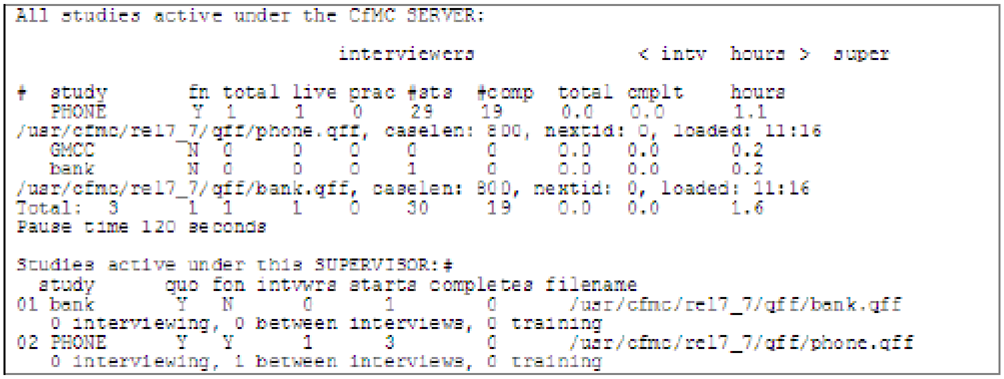
Lists all known (active) studies and whether a study requires a phone file. It will also list studies used by other servers. Entering just DIS displays information on all studies. DIS <study> (i.e., DIS JOB123) will display information on just that study.

The syntax for displaying information on studies is:

```
DIS or DIS <study> or DIS <studyname pattern>
```

The DIS command also does pattern matching on study names. By entering “DIS D*” you will see all the studies that start with a “D” You can use “#” to match numeric digits and “?” to match numbers for alpha characters.

Here is a sample screen and explanation of the information displayed:



```
All studies active under the CfMC SERVER:

                    interviewers               < intv  hours >  super

#  study        fn total live prac #sts  #comp  total  cmplt    hours
   PHONE        Y  1     1    0    29    19     0.0    0.0      1.1
/usr/cfmc/rel7_7/qff/phone.qff, caselen: 800, nextid: 0, loaded: 11:16
   GMCC         N  0     0    0    0     0      0.0    0.0      0.2
   bank         N  0     0    0    1     0      0.0    0.0      0.2
/usr/cfmc/rel7_7/qff/bank.qff, caselen: 800, nextid: 0, loaded: 11:16
Total:  3       1  1     1    0    30    19     0.0    0.0      1.6
Pause time 120 seconds

Studies active under this SUPERVISOR:#
  study      quo fon intvwrs starts completes filename
01 bank       Y   N     0      1        0       /usr/cfmc/rel7_7/qff/bank.qff
   0 interviewing, 0 between interviews, 0 training
02 PHONE      Y   Y     1      3        0       /usr/cfmc/rel7_7/qff/phone.qff
   0 interviewing, 1 between interviews, 0 training
```

where:

study the study name

quo Y/N indicates whether or not a quota file is used with this study (Every shared files study has a quota file).

fon Y/N indicates whether or not a phone file is use with this study.

#ints number of interviewers currently active

#sts number of interviews started so far

#comp number of completed interviews so far

total intv hours total time on interviews by all interviewers so far

total cmplt hours total time on completes by all interviewers so far

super hours total time supervisor has been active

Total Sum of values for all studies and how many studies

DIS does pattern matching on study names, as well. If you type DIS D* you will see only the studies that start with a "D". You can use "#" to match numeric digits and "?" to match numbers or alpha characters.

The "DIS" display has been reorganized, duplicate information has been removed and more information is added. There is a summary total line below the list of studies under the supervisor and server. It prints total number of studies, total interviewers, # starts, # completes, total time on interviews, total time on completes, and total supervisor time.

Specifying LP or LP! at the end of the command will send a copy of the screen to the file indicated on a previous LPDEV command.

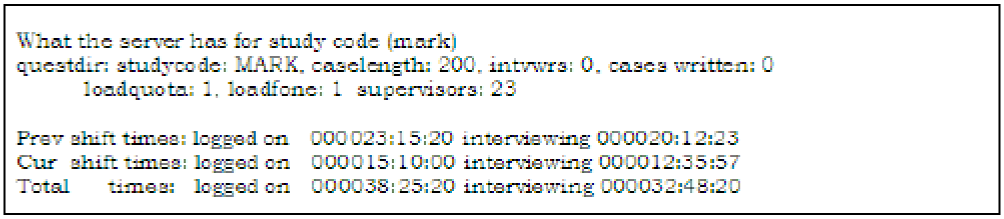
The default DIS screen is updated every 15 seconds with new interviewer information as long as DASHBOARD:NO is not set in your PARMFILE. Use the SURVSUPR command >DUMP a6 to disable this, and >DUMP a7 to enable it again.

STUDY Displays information from the studies file (STUDIES6), from the supervisor's own storage, and from the server about the specified study. It displays the same information as the DIS command, plus total time information.

The syntax for STUDY is:

```
STUDY <study>
```

Here is the information listed in addition to the information that matches the DIS command:


```
What the server has for study code (mark)
questdir: studycode: MARK, caselength: 200, intvwrs: 0, cases written: 0
       loadquota: 1, loadfone: 1  supervisors: 23

Prev shift times: logged on   000023:15:20 interviewing 000020:12:23
Cur  shift times: logged on   000015:10:00 interviewing 000012:35:57
Total      times:  logged on   000038:25:20 interviewing 000032:48:20
```

It also includes load information on the “dummy” interviewer if you are using the EIS,DUMMY option to have phone records returned to Survent to be statused when a connect is not made by the dialer.

PRINT_ALL Print all info for a study, including the QSS screen(s), SPI, DAI and DIS screens; the output goes to the file specified on the LPDEV command. Here is the syntax:

```
PRINT_ALL <study>
```

NOTE: If the LPDEV parameter has not been specified, the output for this command will flash by on the computer screen.

LPDEV Print screens to this device or file Enables you to print screens to a file or printer from QSS, DAI, DIALSTAT, DIS, SPI, or STUDY commands that end with LP, or PRINT_ALL commands.

LPDEV OFF turns off the sending of information to that file or device. Here is the syntax and an example:

```
LPDEV <name/$name.ext/*name/#### LP!/OFF> <PRINTER=name>

      EX:
      LPDEV screens
      SPI bank LP
      LPDEV OFF
```

This would print the output of the SPI command for study bank to the file screens.prt.

“LPDEV OFF” would close the file screens.prt so you could look at or print it.

If you just specify a name it prints name.prt. To print to other names you need to precede the filename with a “$”, for instance, LPDEV $save.me would print to the file save.me.

You can only print directly to a printer in Unix. To print to a printer in DOS, you must print to a file and then print that file.

To direct the output to a printer in UNIX, use the syntax:

```
LPDEV <name> PRINTER=<print program name>
```

"Name" is a file to print to, and "print program name" is the name of the program to use to print the file. The print program name would be lp in SCO Unix and lpr on Linux, and it may be either on other Unix systems.

Using LP on any of the QSS, DAI, DIALSTAT, DIS, SPI, or STUDY command, says to print to the file indicated on the previous LPDEV command and keep the file open for further output. Using LP! says to write the information and then close that file immediately (typically so that file could be printed or read right now).

If you don't specify LPDEV <name>, commands using the LP parameter will send their output to a file named LPSTUFF.PRT.

DIALER-RELATED

These commands are only used if you have an automatic phone dialer connected to your system.

CfMC currently supports SER, PRO-T-S, Stratasoft, and Noble dialers. Contact CfMC for more information.

ATM Start stations under dialer

This starts stations with the same syntax as STS, but connects the interviewers to the dialer when they start.

ATMV Start stations under dialer in "validation" mode

This starts stations in 'validation' mode, where the call gets transferred to a 'validator' at the the end of the survey along with the sample and data records. When the questionnaire executes a "!phone,x" statement, the call and phone record are transferred here at the end of their call. When the validator is done with the call the data and phone records are returned to their files.

DIALER_STATUS Tell current dialer status

This tells you which dialers are running and how many interviewers are running under it.

SERVER:CLEARSTUDYNOW Release a study from dialer

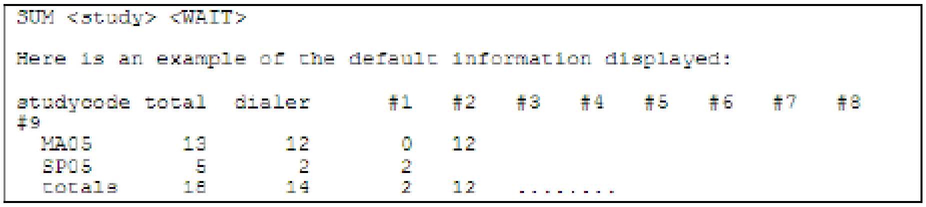
This clears the study and releases it from the dialer even if dialer communications are down. The syntax is:

```
SERVER:CLEARSTUDYNOW <studyname>
```

This does the same as SERVER:CLEARSTUDY except does the additional work to release the study from dialer interactions and logging off the server dispatcher if you are using it.

SUMMARY Show numbers by study under dialers

By default, this displays the total number of users on a specific study, how many are on a dialer and how many are on which dialer if you have multiple dialers. The syntax for SUM is:



```
SUM <study> <WAIT>

Here is an example of the default information displayed:

studycode total  dialer      #1   #2   #3   #4   #5   #6   #7   #8
#9
  MA05        13      12      0   12
  SP05         5       2      2
  totals      18      14      2   12   ........
```

You can modify the display so that a name appears at the top of the headings instead of #1, #2, etc. for the dialers. Do this by adding a five-character name to the end of the DIALER##: line for each dialer in the PARMFILE (See *4.4.5 DIALER RELATED PARMFILE OPTIONS*).

If you specify SUMMARY <study>, you get a list of the live stations running <study>, with their interviewer ids and supervisors <study>:

```
EX:
station(05), supervisor(55), id(DBUG)
```

If you specify the “WAIT” parameter, you get the same list, but it only shows those stations that are in “WAIT” mode. You may then use the “START <#ldev list> <studyname>,WAIT” command to start those stations.

SERVER:DIALER#:WIPEOUT Totally disconnect from dialer Used to reset and restart a dialer. It kills the ATMSREAD (and DIALWRIT if it exists) processes and clears all dialer information for that dialer. Using this, you should be able to restart a dialer from a s supervisor station without taking the server down.

SERVER:>DUMP N Make the CfMC server ignore time differences with the dialer Otherwise, you must synch the time on the CfMC server and dialer computers so that calltimes are correct.

DATA-RELATED

MID Modify the “next case ID” assignment

Displays and allows you to modify the next case ID value to assign for a given study.

Here is the syntax and an example for modifying a case ID:

```
MID <study>

EX:
MID BANK
Present case ID for BANK is 0093
Enter new case ID -->
```

Pressing **Enter** will cause the current case ID to remain unchanged. Entering a number followed by **Enter** will cause that number to become the current case ID value. Changing the case ID could be useful in distinguishing the waves of a study. If you enter a new case ID, it must be greater than the last one used.

Depending on the setting of the MODIFY_CASE_ID header option, you may be prevented from changing the ID, or you may be prompted to verify the change.

VIEW View completed interviews

The “VIEW” subsystem can be invoked from the supervisor to look at completed cases on a study. The syntax is:

```
VIEW study
```

You will be prompted for what case to view and which questions. Responses can be altered if it is allowed. Note that you can put a password on any study that will be prompted for when the supervisor tries to view the study.

For a complete discussion of View mode, see the Survent “View” command above.

COPY Copies the data file of an active study

Causes a copy of the data file for the study specified to be created. An active data file, one from a study that has not been shut down (SDS), is not generally available for use. Active data files are only available in Share mode; they can be read from but not written to by other processes. The COP option facilitates analysis on a currently active study. Here is the syntax and an example for copying a data file:

```
COP <study> <newname>

EX:
COP BC789 BCCOP
```

<Study> is the study name and <newname> is the name of the to-be-created copy of the data file. The new file will have an automatic extension of TR. You can use fully qualified file names. UNIX users see also the SERVER:FASTCOPY command below for faster copies of large files.

NOTE: The COPYFILE utility can be used for this function as well as having other copying options.

SERVER:FASTCOPY (UNIX) Creates a copy of a study data (^tr) file using system commands, which increases the copy speed by a factor of 50 or so. The syntax is:

```
SERVER:FASTCOPY <studyname> <output datafile name>
```

The output datafile name may be a fully qualified name to put the file in other directories. Do not include the .tr extension on the filename as that will be provided automatically. You can use this to copy large files quickly.

SUPERVISOR SETUP

There are many commands designed to control the way the supervisor and interviewers work with the server. The supervisor related setup commands would be put into the file "suprinit" located in the CfMC support directory (eg. cfmc/support). See the file "suprexam" in that directory for examples. Here are the commands you may want to specify in the suprinit file.

Interviewer Setup (most likely only in Suprinit file)

Command/shorthand	**Description**
TIMEOUT	Set time un-started stations dropped
FORCE_INTO_INTERVIEW (FORCEII)	Idle time between interviews after which interviewer is forced into another interview
FORCE_INTO_BREAK	Idle time between interviews after which nterviewer is forced into break mode
FORCE_OUT_OF_SESSION (FORCEOOS)	Idle time between interviews after which interviewer is forced to quit session
END_SHIFT (ENDSHIFT)	Reset intrvwer stats to 0 for DAI STATS

Supervisor Setup (most likely only in Suprinit file)

Command/shorthand	**Description**
PAUSE	Set time to stay in command mode
WAIT	Set pause between screens in monitor
DISALLOW	Disallow certain modifying keywords
SAVE_SELECT	Save Hide/Rev/List criteria for reuse?
MAX_SESSION_SINGLE_START (MAX_SESSION_SS)	Limit #stations on one START command
END_MON_SURVENT (ENDMONSURV)	Do a review interview after each monitored interview?

Supervisor Automatic Command-Scheduling Commands

Command/shorthand	**Description**
AT	Say when to do command
EVERY	Say howoften to do command
SHOW_TIMED_COMMANDS (SHOWTIMED)	Show current scheduled commands
DELETE_TIMED_COMMAND (DELTIMED)	Delete a scheduled command
SUSPEND_TIMED_COMMAND (SUSPTIMED)	Suspend a scheduled command
ACTIVATE_TIMED_COMMAND (ACTTIMED)	Reactivate scheduled command

INTERVIEWER SETUP

TIMEOUT Sets the time after which idle “started” interviewers will be dropped

This lets you specify how long in minutes to leave a station in start mode waiting for an interviewer before logging off the station. The default is 30 minutes; the range of values can be 0-240. If you do not specify a value, it just reports the current setting.

If the value is set to 0 (zero), it means "never timeout." In this case, the interviewer will never be removed from the system for being idle. The syntax for TIMEOUT is:

```
TIMEOUT <###>
```

FORCE_INTO_INTERVIEW Forces interviewers into new interview after # seconds

This causes interviewers at the "Return or Q to Quit" prompt into the first screen of the interview after # seconds. If an interviewer is started by a supervisor after FORCE_INTO_INTERVIEW=60 is specified, they will be placed on the first question of the interview after 60 seconds if they have not pressed the **Enter** key to start before then.

The syntax is:

```
FORCE_INTO_INTERVIEW=## LOG
```

The default is 0, which means the interviewer may stay at the **Return** to quit prompt for as long as they like. If the number specified is negative, the number will be used, but no message will be sent to the supervisor.

The two most useful aspects of this feature are:

1 Interviewers cannot stay at the Return to Quit prompt for excessive amounts of time. If you have a dialer, the interviewer will made available for calls as soon as this goes into affect.

2 It facilitates time logging, in that you can set a timer in the interview to record the amount of time before a phone number is actually called or an interview starts.

If you add ",LOG" after the number of seconds, it will be logged in the Server log (LL) file. This command would most likely be place in the SUPRINIT file so it will be set every time a supervisor logs on.

You can now use the option to specify how many seconds the respondent can stay at the “between interviews” prompt before being forced either into the interview or out of their session. This was previously only controllable at the supervisor for all of their stations. If the supervisor specifies a time for either of these options, it will override what the questionnaire dictates.

As for the supervisor option, a "-" before the number means to not tell the supervisor every time this happens.

You can also say "FORCE_INTO_INTERVIEW=-5:LOG and it will log the information to the server log file when it forces interviewers into interviews. This applies to "FORCE_OUT_OF_SESSION=#",AND "FORCE_INTO_BREAK" also. (Those options follow.)

NOTE: The "force into interview" option could previously be done using !SPC,H,A,## which can still be used.

FORCE_INTO_BREAK Forces interviewers into “break” mode after # seconds

This causes the interviewer to be forced into “break” mode if they have not started a new interview before # seconds. If the number is positive, a message will be sent to the supervisor; if the number is negative, no message will be sent. Adding “,LOG” after the number of seconds will cause it to be logged in the Server log (LL) file.

This command is used so that reports reflect the proper time and the interviewers don’t appear as active on the supervisor’s interviewer activity screens.

FORCE_OUT_OF_SESSION Forces interviewers out of session after # seconds

This causes the interviewer to be forced out of their interviewing session if they have not started a new interview before # seconds. If the number is positive, a message will be sent to the supervisor; if the number is negative, no message will be sent. Adding “,LOG” after the number of seconds will cause it to be logged in the Server log (LL) file.

For **webCATI**, this option will force someone out of an interview if on a question for more than a # of minutes. This is useful if you want the questionnaire to end if the respondent goes away for a long period of time. This value must be less than the "suspend_timeout" time specified in order for it to work.

This command is used to log inactive interviewers off so that reports reflect the proper time and the interviewers don't appear on the supervisor's interviewer activity screens.

END_SHIFT Sets counters to 0 for shift totals

This marks the end of shift for DAI STATS screens "SHIFT TOTALS" and resets them to 0.

SWITCHLOG Causes the server to close the current log file and start a new one.

It does the same thing as "SERVER:LOG". This lets users disallow "SERVER:" commands and still switch the logs.

Supervisor Setup

PAUSE Set keyboard timeout at command prompt

Lets you change the automatic keyboard timeout to give you more time to enter commands. The default value is 120 seconds; you can enter any positive number from 10 to 300. Here is the syntax and an example for changing the pause rate:

```
PAUSE <seconds>

EX:
PAUSE 60
```

Furthermore, if you want to pause more than 5 minutes, you can say, for instance, PAUSE 500#. The "#" will allow you to say any number you want.

WAIT Set the delay between questions when monitoring

Lets you delay the refreshing of the supervisor's screen when monitoring. The default is

10 tenths of a second (1 second). The allowable range is 5 to 100. Here is the syntax and

an example for monitor delay:

```
WAIT <tenths of second>

EX:
WAIT 50
```

This command only controls nonTEX questions. For TEX questions, the text is sent every 20 characters typed.

This would typically be placed in the SUPRINIT file so that it activates each time a supervisor logs on.

DISALLOW Disallow supervisor commands

This will disallow particular commands from the supervisor. The syntax for DISALLOW is:

```
DISALLOW <command>
```

The command DISALLOW QSS MOD will disallow modifying quotas, but will not disallow looking at quotas.

This would typically be placed in the SUPRINIT file so that it activates each time a supervisor logs on. Once commands are disallowed, they cannot be allowed during that supervisor session. You must restart the supervisor without the DISALLOW command on.

SAVE_SELECT Save SELECT statements used in HIDE or REVEAL to be used later This command saves each SELECT statement used in a file in the sub-directory study.s_ in the supervisor's directory with the name C# with # being a number from 1 to 9999.

When using this option, the prompt for a HIDE or REVEAL will read as follows:

```
Select filename: \cfmc\super\exam2.s_\C001

Contents: [1.3#415]
T_ake it, N_ext, or Quit-->
```

T will use that Select

N will look for the next available Select

Q will go to the "Enter SELECT statement or ALL" prompt.

If used, this is typically placed in the SUPRINIT file to be set whenever a supervisor logs on.

MAX_SESSION_SINGLE_START Limits range of stations started concurrently

This command keeps supervisors from trying to start stations that don’t exist or more stations than they expected to. The syntax is:

```
MAX_SESSION_SINGLE_START #,#
```

There are two values. The first is for max ever and the second for the number (#) above which you are warned that you may be starting too many. The default maximum, if not specified, is 100, and the default warn value is 50. This means you can only start 100 interviewers maximum on a particular study at one time. And, you can only start up to 50 without a warning. This command would most likely be placed in the SUPRINIT file as a start-up parameter.

END_MON_SURV Causes a Survent interview to run after each monitored interview

This lets you run Survent after each monitored interview to “rate” the monitored interviewer. If this is used, the supervisor starts a Survent interview after each monitored interviewer to record their rating of the interview session. The program automatically provides certain information to the interview, like the start time, study name, interviewer id, monitor id, etc. You must write a custom questionnaire called to ask the questions of the monitor. The syntax is:

```
END_MON_SURV <interviewer id>
```

The interviewer ID is used to name the questionnaire to use and data file that will be created with the responses to the rating interview; if the interviewer id is “MON1”, the questionnaire used will be “MON1^QFF” and the data file created will be “MON1^TR”.

“ENDMONSURV” without any parameters turns off the execution of interviews after each monitored survey. See the SURVMON program for complete information.

Miscellaneous Setup commands and Dump Switches

There are some commands that are seldom used but need to be documented so here they are. There are also many "dump switches" in CfMC software that are not documented, but here is a list of switches that may help your operations:

SERVER:TASKDELAY Delays HIDE/REVEAL operations to limit CPU usage

This sets a delay in milliseconds for batch operations such as HIDE, REVEAL, CHANGE_OWNER, and COPY. This will keep those tasks from slowing down your system too much. The delay will be after every five cases read for the task. Here is the syntax and an example for SERVER:TASKDELAY:

```
    SERVER:TASKDELAY <milliseconds>

    EX:
    SERVER:TASKDELAY 20
```

>Dump 01	Overrides confirmation of SERVER: commands
>Dump a6	Disables the dashboard (self-updating) screens
>DUMP a7	Enables the dashboard (self-updating) screens
>DUMP 05	Suppresses "Out of Numbers" message to the supervisor
>DUMP s	Shows all messages to and from the server for debugging problems.
>DUMP y2	Overrides "case not found" message when using SPC,P

Timed Commands

You can specify that commands be run automatically at certain times under the server using the SURVSUPR commands AT and/or EVERY to tell the program when to process the commands.

AT says to do the command one time only at some later time. The syntax is:

```
AT <time> <command>
```

The <time> may be any valid time supported in the CfMC software. The <command> may be any valid SURVSUPR command. Note that the <command> may be an external file reference such as "&suprcomd" to read a set of commands to be executed.

```
EX:
AT mon 3:30pm QSS bank
```

EVERY says to do the command at regular intervals as long as that particular SUPERVISOR is running under the server. As with the AT command, the <command> may be an external file reference to read a set of commands to be executed. The syntax for the various formats of the EVERY command is:

```
EVERY [#] MINUTES <command>
EVERY [#] HOURS <command>
EVERY HOUR_ON_THE_HOUR <command>
EVERY STARTUP <command>
```

EVERY 10 MINUTES SPI FRED will do the SPI command on study FRED every 10 minutes. EVERY HOUR QSS J123 will do a quota display for job J123 every hour starting now. EVERY HOUROTH DIS will do the DIS command every hour on the hour.

EVERY STARTUP will do the <command> at every supervisor startup from now on.

AT and EVERY write commands to a file cfmt<ldev> in the CFMCCFG (UNIX) or IPCFILES (DOS) area. This file is read (if it exists) at supervisor startup time and startup commands are executed.

This command applies to a particular Supervisor only.

SHOW_TIMED_COMMANDS lets you view the existing timed commands. This will display a list of the scheduled commands and their command number, which can be used in the following commands.

```
EX:
SHOWTIMED
```

DELETE_TIMED_COMMAND # lets you override a current timed command.

```
EX:
DELTIMED 2
```

SUSPEND_TIMED_COMMAND # lets you temporarily suspend a command.

```
EX:
SUSPTIMED 3
```

ACTIVATE_TIMED_COMMAND # lets you reactivate a suspended command.

```
EX:
ACTTIMED 3
```

Writing Command Files

You can also write your own command files which can be used to automatically do procedures that would otherwise require a lot of work for the supervisors. See below for more information on how to write command files and set up defines.

The rules for writing command files in SURVSUPR are slightly different than in other programs.

- Commands start processing after the keyword Begin and will continue to attempt to process each line as a command until the keyword End is encountered.
- Each command must be on its own line.
- Commands that require further interaction will pause and ask for the needed information before continuing.

Command files are intended to help automate a large interviewing shop which would more efficiently operate by having a programmer enter the commands in a file and then allow supervisors to read the file into SURVSUPR with a simple command like “&start.spx”.

Here are some examples of command files:

```
File START.SPX -
BEGIN
PAUSE 120
STS 1-20 study1
STS 21-70 study2
END
File STOP.SPX -
BEGIN
STP 1-20
STP 21-70
SDS study1
SDS study2
SERVER:DOWNNOW
QUIT
END
```

NOTE: If you write a command file to execute commands on a study with a password, you can specify the password on a line by itself after the command requiring the password and it will be able to execute the command. If you wish, you can override the password protection temporarily by using the command ">DUMP O3" at the beginning of your spec file. Use ">DUMP -O3" at the end of the spec file to turn password protection back on.

4.4.3 The CfMC SERVER

To run the CfMC SERVER, enter:

```
SERVER ### (DOS/UNIX)
server bg (UNIX, "BG" to run in the "background")
```

The CfMC SERVER program delivers and retrieves data from the different components in the SHARED FILES interviewing environment. All processes communicate through the SERVER. If a SURVSUPR session wants to see the current quota values, it does not go directly to the quota file and get the information. Instead, a request for the SERVER to retrieve the information from the quota file is given by SURVSUPR. When a Survent process has finished with an interview, the Survent process requests that the SERVER pick up the data case and deliver it to the end of the data file.

You send commands to the server via the SURVSUPR program. Here is a list of some of the commands to the server that you may type in the SURVSUPR program:

SERVER: DOWN Stops the SERVER. If interviewers are still running, it will not go down.

SERVER: DOWNNOW Forces the server to go down even if studies are still running,

SERVER: CLEARSTUDY <studyname> Clears all access to study files of the indicated study. Try the SDS and DOWNSTUDY commands before attempting this.

SERVER: CLEAR <terminal ID list> Clears all terminal i/o to the station, and clears the station entry out of the server's stations file. This should only be used after attempting to STOP or KILL the station using supervisor commands.

SERVER: LOG <filename> Closes the current server log file and starts a new one. If you specify a name it creates the new file with that name. This is useful to look at a log file immediately after a problem has occurred without taking the server down.

SERVER Commands to Dialers

In the supervisor, you may use SERVER: DIALER:## <command> to send a command to a particular dialer.

Here are some examples:

```
SERVER:DIALER:1 CLOSE Close the dialer and dialer port
SERVER:DIALER:2 INIT Initialize the dialer interface
SERVER:DIALER:3 WIPEOUT Severs all connections to the dialer
SERVER:DIALER:4 MSG:CC:chip Close campaign "chip" at the EIS dialer
```

Server Logging

The server writes fixed format log records to the log file for events on the system. You can use these records to create reports on the number of interviewer starts/stops, number of phone numbers gotten, number of completes, etc. These records will be interspersed with the other log records in the file (named Lx###### in the directory named LOGS below where the SERVER is run). The server stores its log files in the "logs" subdirectory by default. You can control the name of the directory the server writes log files to by using the system variable CFMCLOGS <directory name>, eg. 'setenv CFMCLOGS /usr/cfmc/mylogs'. You can change it save files in the local directory by entering 'setenv CFMCLOGS ./' in UNIX and 'set CFMCLOGS=.\' in DOS. You can control the number of lines to write to each log file with the option "LL_LOGGING:YES <# of lines> specified on the SERVER command. The default is 20,000 lines.

See LGRPT^SPX in CFMC\SURVENT for sample reports created from the server log records.

In UNIX and DOS systems, the CfMC server log file names include the year, month, day, hour, minute and seconds when the file has been created. Previous software versions had other information that is now not included in the format. The name format is:

ll<yyyymmddhhmmss>.<ldev>

Where yyyymmddhhmmss represents the year, month, day, hour, minute, and seconds.

Example:

```
     ll20041201153223.9901
```

This filename would have been created on December 1, 2004 at 3:32:23 pm for device number 9901.

The format of the records is as follows:

```
LL## nnnnyymmddhhmmssdjjjssssssss iiii
1-4: LL##   ## (currently 03-12) is the type of log record
5:          blank
6-9: nnnn   device (station) number
10-25: yy...year, month, day of month, hour (military time),
minute, second, day of week (1=Monday), and julian date
26-33: ss...study code
34:   P (if in Practice Mode)
35-38: iiii interviewer ID
39:         blank
```

The format for the rest of the record (starting in column 40) is:

Log type 01: Start times for supervisors, monitors and mentor processes

```
EX:
LL01 0043010918162036226l(none) **** 1 survsupr start
```

This allows you to report on supervisors and monitors in a similar manner to interviewer reporting. Another application would be to keep track of the length of Mentor processes. The record formats are the same, except one says **"start"** where the other says **"stop."**

Log type 02: Records stop times for these processes.

```
EX:
LL02 0043010918162636226l(none) **** 1 survsupr stop
```

Format of LL01/LL02 records:

```
1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
(none)   **** p <programname sssss>
```

Where:

p - program number (1=survsupr/2=survmon/3=Mentor)

<programname> - <survsupr/mentor/survmon sssss -"start" (LL01) or "stop" (LL02)>

Log type 03: Interviewer start

Format:

```
      1 |       2         3         4         5         6         7         8
      12345678901234567890123456789012345678901234567890123456789012345678901234567890
                      sssssssss iiii eeee cccc A P (sssssssss) bbbb R I

35-38: iiii           interviewer id
40-43: eeee           phone extension
44:                   blank
45-48: cccc           sound server channel
49:                   blank
50:     A             whether using dialer (0=no, 1=predictive, 2=preview mode)
51:                   blank
52:     P             practice mode (1=yes or 0=no)
53:                   blank
54-64: (sssssssss)    special type (1-9,0)
65:                   blank
66-69: bbbb           ldev of boss, 0 if no boss
70:                   blank
71:                   reason for stop
                          K: killed by Supervisor
                                N: interview stopped normally
                                S: cleared by Supervisor or Server
                      type of start
                              R-B: Back from break/lunch
                                I: Initial Startup (Netsurv or Supervised)

72:                   blank
73:     I             Interview type:
                      Supervised(B), (W)ebcati, Web(S)urvent, (C)ati
```

```
EX:
LL03 00660110151515181288ODBC DBUG 0000 0000 0 0 (0 ) 0000 I
```

Log type 04: Interviewer stop

Format:

```
Sssssss iiii bbbb yymmddhhmmdjjj sssss ccccc RAFnnnnn

26-33   ssssssss    study code
34:                 blank
35-38:    iiii      interviewer ID
39:                 blank
40-43:    bbbb      ldev of supervisor, 0 if no supervisor
44:                 blank
45-58:    yymmdd... date/time this interviewer started*
59:                 blank
60-64:    sssss     # interview starts*
65:                 blank
66-70:    ccccc     # cases written*
71:                 blank
72:       R         reason stopped
                    B: gone on break
                    C: Cleared by "Clearstudy" command
                    D: Cleared by "server:clear" command
                    F: forced to quit with Force_Out_Of_Session
                    K: killed by supervisor
                    L: went to lunch
                    M: went to meeting
                    N: normal quit
                    S: killed self
                    X: Changed to new study by CHI command
                    W: killed by supervisor (kill #123)
                    Z: cleared by supervisor (clear #123)
73:       A         dialer_type (1=SER, 2=MSG/PREDICTIVE, 4= NOBLE, 8=STRATASOFT)
74:       F         study used a phone file (1=yes or 0=no)
75-79:    nnnnn     # of phone records with histories added this session
```

```
EX:
LL04 0959981209084544334 3Main  Q177 0000 199812090825023343 00029 00000S0100
```

*These fields are now "since the corresponding LL03 record" instead of "since the beginning of the session."

Log type 05: Interview start (actually get quota values)

```
Format: ssssssss iiii  mmmm r d v

26-33:    ssssssss study code
35-38:    iiii     interviewer id
40-43:    mmmm     ldev of monitoring device
44:                blank
45:       r        "1" if RDG mode interview, otherwise "0"
46:                blank
47:       d        "1" if debug mode interview, otherwise "0"
48:                blank
49:       v        "1" if view mode interview, otherwise "0"
```

```
EX:
LL05 09599812090845443343MAIN Q177 0000 1 0 0
```

Log type 06: Interview end (actually put case)

```
Format: cccccccccc A

40-49:    ccc...    case ID, left-justified
50:                 blank
51:       A         1: case added to file.
                    0: case replaced in file.
```

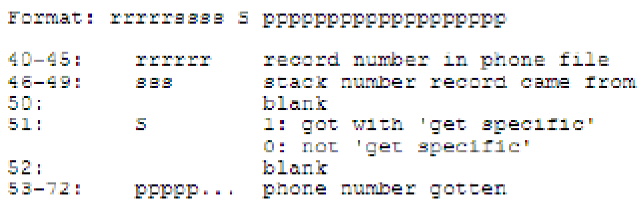
```
EX:
LL06 00039507131725134194 EXAM1 BKOK 0022 1
```

Log type 07: Interview complete/abort; view mode; backups, consecutive backups

```
Format: ssssssss iiii c m bbb BBB v xxxxxxxxxx


26-33:      ssssssss study code
35-38:      iiii     interviewer ID
40:         c        whether completed or aborted:
                     0: abort
                     1: interview completed
                     2: interview suspended
41:                  blank
42:         m        whether monitored
                     0:      interview was never monitored
                     1:      interview was monitored at some point
44-46       bbb      Number of backups
48-50       BBB      Number of consecutive backups
52          v        1, if viewed, 2, if viewed and altered
53-63       xxxxxxxxxx case ID, if viewed
```

Log type 11: Get phone number.

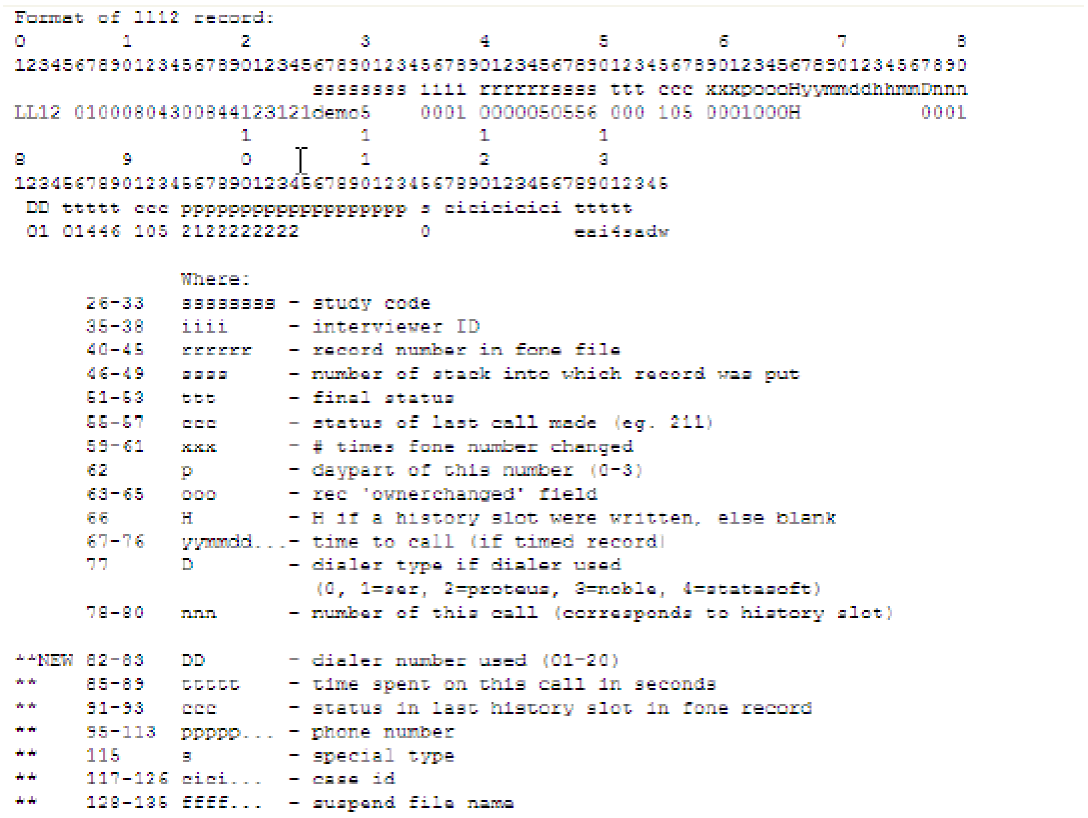
EX:



```
Format: rrrrrssss 5 ppppppppppppppppppp

40-45:    rrrrrr    record number in phone file
46-49:    sss       stack number record came from
50:                 blank
51:       5         1: got with 'get specific'
                    0: not 'get specific'
52:                 blank
53-72:    ppppp...  phone number gotten
```

```
LL11 0003958713172423419 4 EXAM1      BKOK 0001210385 0 4155329727
```

Log type 12: Put phone number, phone status, history, dialer number and dialer switches



```
Formet of 1112 record:
0         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
                         ssssssss iiii rrrrrrssss ttt ccc xxxpoooHyymmddhhmmDnnn
LL12 01000804300844123121demo5    0001 0000050556 000 105 0001000H          0001
                   1         1         1         1
8         9        0         1         2         3
12345678901234567890123456789012345678901234567890123456
 DD ttttt ccc ppppppppppppppppppp s cicicicici ttttt
 01 01446 105 2122222222          0           esi4sadw

               Where:
       26-33   ssssssss - study code
       35-38   iiii     - interviewer ID
       40-45   rrrrrr   - record number in fone file
       46-49   ssss     - number of stack into which record was put
       51-53   ttt      - final status
       55-57   ccc      - status of last call made (eg. 211)
       59-61   xxx      - # times fone number changed
       62      p        - daypart of this number (0-3)
       63-65   ooo      - rec 'ownerchanged' field
       66      H        - H if a history slot were written, else blank
       67-76   yymmdd...- time to call (if timed record)
       77      D        - dialer type if dialer used
                          (0, 1=ser, 2=proteus, 3=noble, 4=statasoft)
       78-80   nnn      - number of this call (corresponds to history slot)

**NEW 82-83   DD       - dialer number used (01-20)
**     85-89   ttttt    - time spent on this call in seconds
**     91-93   ccc      - status in last history slot in fone record
**     95-113  ppppp... - phone number
**     115     s        - special type
**     117-126 cici...  - case id
**     128-136 ffff...  - suspend file name
```

Log type 13: more info when put phone number

```
Format: ttttt ccc pppppppppppppppppp

40-44:    ttttt     time spent on call in seconds
45:                 blank
46-489:   sss       status in last history slot in phone record
50:                 blank
51-68:    pppppp…   phone number put back
70-79:  yymmddhhmm  time to call back
```

Log type 14: User info from interview

```
Format: mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm

40-79:    mmmm…     Message from SPC,E,3 statement
```

This places up to 40 columns of information starting in column 40 that is specified on the !SPC,E,3 statement text line. This is useful to have your own logging system with information that the CfMC system doesn't otherwise collect.

Log type 15: returns record when !phone,9 is executed

```
EX:
LL15 004302010318254840003PHONE DBUG at phone,9
```

This is useful because it usually indicates the beginning of the actual questionnaire for reporting purposes, so you can separate out time before and after it is executed when reporting interview times. It reports the station #, time, study and interviewer ID.

Log type 16: time between questions

```
Format: time on qq: qstlabel ###

Where:

40-50:                "time on qq:"
52-59: qqqqqqqq       question label
61-66: nnnnnn         time on question in seconds
68-74                 "seconds"
```

```
EX:
LL16 004302010318254840003PHONE DBUG time on qq:AGE    15 seconds
```

This is only recorded if the "time_questions" keyword is specified in the questionnaire header for a particular study.

Log type 17: when a phone number is sent to the dialer.

This enables you to track numbers better. The record has the standard information in columns 1-39 and specific information in columns 40-70. The format of the log record is as follows:

```
Format:
4         5         6         7         8
01234567890123456789012345678901234567890
rrrrrrssss S ppppppppppppppppppp


40-45: rrrrrr Record number retrieved from fone file
46-49  ssss   Stack number record in fone file came from
50:    blank
51:    S How sent: 0: Requested by dialer 1: got with 'get specific'
52-70  ppp... - phone number sent
```

Log type 18 logs quota updates.

All quota updates on the LL18 record in the server log files are recorded. The format is the same for the first 39 columns as other "LL" records. Here is the format starting in column 40:

```
4         5         6         7         8
0123456789012345678901234567890123456789012345

<- quota name               -> t cccccc vvvvvv
```

If the quota is a numbered quota, the “quota name” will be “# “. “t” is the quota type (I=increment, S=set to value), “c” is the quota increment, and "v" is the new value.

Survent, Supervisor logging

The server log files are now saved in a file named XXX ll. (year, month, day, hours, minutes, seconds.) Note that you cannot turn logging off).

“LOGGING:" can be used instead of "LL_LOGGING:" as well.

If you want to log the supervisor or Survent, you can by specifying "LOGGING: SURVSUPR SURVENT". "LOGGING: NO" will cause the Survent and Survsupr not to be logged (default). The files created when logging Survent are "sv" Survsupr log files start with "su".

Summary Interviewer Log file

The server will also log interviewing sessions to a summary file named "intvrlog.yyyy_mm_dd" (where yyyy_mm_dd is the date that file was created (year_month_day)). It creates a record for each interviewing session, and updates it at the end of each interview. The file is created in the CfMC IPCFILES directory. It has all the information below about that interviewing session in it. A new file is created at the start of each day a study is started.

This enables users to get simple reports on interviewer times without reading through all of the server log files. There are standard reports that use this file (see CFMC/survent/logrept.spx), or you can write your own.

In DOS, the interviewer logging file is named ‘intvrlog.X##’, where “X” is the month letter (A-L for 1-12) and “##” is the day of the month. In UNIX, the filename is ‘intvrlog.__’ (year/month/day).

The information stored appears as follows:

```
Start Stop    Size Info recorded          Value/Notes
----- ----    ---- -------------          ----------
1    -  4       4 Interviewer ID          From employee.xxx 1-4
5    -  29     25 Interviewer Name        From employee.xxx 6-30
30   -  37      8 Study name              (4-8 characters)
38   -  55     18 Start Time              (yyyymmddhhmmssjjjd)
56   -  69     14 Time of last event      (yyyymmddhhmmss)
70   -  73      4 Device Number           (1-9999)
74   -  77      4 Dialer Extension        (1-9999)
78   -  81      4 Supervisor Device #     (1-9999)
82   -  85      4 SoundServer Ext.        (1-9999)
86   -  86      1 Practice Mode           (Y/N)
87   -  95      9 Special Type Flags      (123456789)
96   -  96      1 Web-Cati                (Y/N)
97   -  97      1 ODBC Interface          (Y/N)
98   -  98      1 How Stopped             (A=Active/Bad Stop
                                          B=Blow Error
                                          C=Cleared
                                          I=Between Interviews
                                          K=Killed
                                          Q=Quit
                                          S=Stop)
99   -  100    2 Dialer number used       (01-09)
101 -  102     2 Site code                (01-99)
103 -  103     1 Dialer type              E (Eis),N (Noble),P
                                          (Msg/PRO-T-S),S (Stratasoft)
104 -  104     1 Dialing mode             0=none, 1=predictive,
                                          2=preview
105 -  105     1 Current Status           I- In interview, After !phone,9
                                          P- At  prompt
                                          B- On break
                                          L- At lunch
                                          W- Waiting for dialer
                                          A- At start of interview
                                          (before !phone,4/5)
                                          F- Got fone record, before
                                          phone,9
                                          M- In Main interview (after
                                          !spc,e,E)
                                          E- At end (after !spc,e,F)
                                          R- In RDG
106 - 106       Currently monitored:      Y/N
```

```
107 - 107      View Mode:              Y/N
108 - 108      Coding mode:            Y/N

Times in Seconds:
121 - 125   5  Connect Time          (1-99999)
126 - 130   5  Between Interviews    (1-99999)
131 - 135   5  Time on Break-paid    (1-99999)
136 - 140   5  Time on Lunch-unpaid  (1-99999)
141 - 145   5  Time Monitored        (1-99999)
146 - 150   5  Total Active Time     (1-99999)
151 - 155   5  Time in 'View' Mode   (1-99999)
156 - 160   5  Time Active-View Mode (1-99999)
161 - 165   5  Time Waiting for Phn  (1-99999) Time waiting for dialer
166 - 170   5  Time Before Phone #   (1-99999) Time before !Phone,4/5 gets
                                               number
171 - 175   5  Time on Completes     (1-99999) Status 1 or CountasComplete
176 - 180   5  Time in Screener      (1-99999) Time until Spc,e,E
181 - 185   5  Time in Main          (1-99999) Time from Spc,e,E to Spc,e,F
186 - 190   5  Time in Review/Edit   (1-99999) Time after Spc,e,F
191 - 195   5  Time on Resolved      (1-99999) Resolved 1-99 not complete
196 - 200   5  Time on Timed Calls   (1-99999) Status 104-105,160-179
201 - 205   5  Time on Timed Notime  (1-99999) Status 105
206 - 210   5  Time on Other Active  (1-99999) Status 102,106-159,180-199
211 - 215   5  Time on No History    (1-99999) Status 201-214
216 - 220   5  Time on Suspends      (1-99999) Suspended Interviews
221 - 225   5  Time on Comp. Resumes (1-99999) Resumes completed
226 - 230   5  Time on Coding        (1-99999) Updating existing cases
231 - 235   5  Time running RDG      (1-99999) Random data generation test
                                               time

Summary Counts:
301 - 304   4  # of interview starts (1-9999)
305 - 308   4  # of completed intvws (1-9999)  Status "1" or CountasComplete
309 - 312   4  # Updated case        (1-9999)  Completed with coding mode
313 - 316   4  # Starts Debug mode   (1-9999)
317 - 320   4  # Starts RDG mode     (1-9999)  Random Data Generation Starts
321 - 324   4  # RDG mode completes  (1-9999)  Random Data Generation
                                               Completes
325 - 328   4  # Starts View mode    (1-9999)
329 - 332   4  # View mode updates   (1-9999)
333 - 336   4  # Regular mode starts (1-9999)
337 - 340   4  # Regular completes   (1-9999)
341 - 344   4  # Regular updates     (1-9999)  Regular updates/coding mode
345 - 348   4  # Screeners done      (1-9999)
349 - 352   4  # Editing/Review done (1-9999)
353 - 356   4  # Interviews monitored(1-9999)
```

```
357 - 360   4 # Completes monitored  (1-9999)
361 - 364   4  # Suspends            (1-9999)
365 - 368   4  # Resumes completed   (1-9999)
369 - 372   4  # Calls made          (1-9999)
373 - 376   4  # Specific #s called  (1-9999)  !Phone,5 executed
377 - 380   4  # Resolved calls made (1-9999)  Status 1-99 but not completed
381 - 384   4  # Timed calls         (1-9999)
385 - 388   4  # Other active calls  (1-9999)
389 - 382   4  # No History calls    (1-9999)
393 - 396   4  # Times changed Ph. # (1-9999)
397 - 390   4  # Owned #s retrieved  (1-9999)  !Phone,5 executed
401 - 404   4  # Changed Ownership   (1-9999)
405 - 408   4  # Special Intv. #1 #s (1-9999)
409 - 412   4  # Special Intv. #2 #s (1-9999)
413 - 416   4  # Special Intv. #3 #s (1-9999)
417 - 420   4  # Special Intv. #4 #s (1-9999)
421 - 424   4  # Special Intv. #5 #s (1-9999)
425 - 428   4  # Special Intv. #6 #s (1-9999)
429 - 432   4  # Special Intv. #7 #s (1-9999)
433 - 436   4  # Special Intv. #8 #s (1-9999)
437 - 440   4  # Special Intv. #9 #s (1-9999)
441 - 444   4  # Breaks              (1-9999)
445 - 448   4  # Lunch breaks        (1-9999)
449 - 452   4  # Times monitored     (1-9999)

# of numbers given each status 1-250:
501 - 1500  4 # with each status    (1-9999)
501 - 504   4 # of Status Code #001 (1-9999) Complete
505 - 508   4 # of Status Code #002 (1-9999) Refusal
509 - 512   4 # of Status Code #003 (1-9999) Language
513 - 516   4 # of Status Code #004 (1-9999) Mid-Terminate
517 - 520   4 # of Status Code #005 (1-9999) Non-Working
521 - 524   4 # of Status Code #006 (1-9999) Non-Residential Number
525 - 528   4 # of Status Code #007 (1-9999) Non-Business Number
529 - 532   4 # of Status Code #008 (1-9999)
533 - 536   4 # of Status Code #009 (1-9999)
537 - 540   4 # of Status Code #010 (1-9999)
541 - 544   4 # of Status Code #011 (1-9999)
545 - 548   4 # of Status Code #012 (1-9999)
549 - 552   4 # of Status Code #013 (1-9999)
553 - 556   4 # of Status Code #014 (1-9999)
557 - 560   4 # of Status Code #015 (1-9999)
561 - 564   4 # of Status Code #016 (1-9999)
565 - 568   4 # of Status Code #017 (1-9999)
569 - 572   4 # of Status Code #018 (1-9999)
573 - 576   4 # of Status Code #019 (1-9999)
```

```
577 - 580   4  # of Status Code #020 (1-9999)
581 - 584   4  # of Status Code #021 (1-9999)
585 - 588   4  # of Status Code #022 (1-9999)
589 - 592   4  # of Status Code #023 (1-9999)
593 - 596   4  # of Status Code #024 (1-9999)
597 - 600   4  # of Status Code #025 (1-9999)
601 - 604   4  # of Status Code #026 (1-9999)
605 - 608   4  # of Status Code #027 (1-9999)
609 - 612   4  # of Status Code #028 (1-9999)
613 - 616   4  # of Status Code #029 (1-9999)
617 - 620   4  # of Status Code #030 (1-9999)
621 - 624   4  # of Status Code #031 (1-9999)
625 - 628   4  # of Status Code #032 (1-9999)
629 - 632   4  # of Status Code #033 (1-9999)
633 - 636   4  # of Status Code #034 (1-9999)
637 - 640   4  # of Status Code #035 (1-9999)
641 - 644   4  # of Status Code #036 (1-9999)
645 - 648   4  # of Status Code #037 (1-9999)
649 - 652   4  # of Status Code #038 (1-9999)
653 - 656   4  # of Status Code #039 (1-9999)
657 - 660   4  # of Status Code #040 (1-9999)
661 - 664   4  # of Status Code #041 (1-9999)
665 - 668   4  # of Status Code #042 (1-9999)
669 - 672   4  # of Status Code #043 (1-9999)
673 - 676   4  # of Status Code #044 (1-9999)
677 - 680   4  # of Status Code #045 (1-9999)
681 - 684   4  # of Status Code #046 (1-9999)
685 - 688   4  # of Status Code #047 (1-9999)
689 - 692   4  # of Status Code #048 (1-9999)
693 - 696   4  # of Status Code #049 (1-9999)
697 - 700   4  # of Status Code #050 (1-9999)
701 - 704   4  # of Status Code #051 (1-9999)
705 - 708   4  # of Status Code #052 (1-9999)
709 - 712   4  # of Status Code #053 (1-9999)
713 - 716   4  # of Status Code #054 (1-9999)
717 - 720   4  # of Status Code #055 (1-9999)
721 - 724   4  # of Status Code #056 (1-9999)
725 - 728   4  # of Status Code #057 (1-9999)
729 - 732   4  # of Status Code #058 (1-9999)
733 - 736   4  # of Status Code #059 (1-9999)
737 - 740   4  # of Status Code #060 (1-9999)
741 - 744   4  # of Status Code #061 (1-9999)
745 - 748   4  # of Status Code #062 (1-9999)
749 - 752   4  # of Status Code #063 (1-9999)
753 - 756   4  # of Status Code #064 (1-9999)
757 - 760   4  # of Status Code #065 (1-9999)
```

```
     761 - 764   4  # of Status Code #066 (1-9999)
     765 - 768   4  # of Status Code #067 (1-9999)
     769 - 772   4  # of Status Code #068 (1-9999)
     773 - 776   4  # of Status Code #069 (1-9999)
     777 - 780   4  # of Status Code #070 (1-9999)
     781 - 784   4  # of Status Code #071 (1-9999) Dialer-Number Not Dialed
     785 - 788   4  # of Status Code #072 (1-9999) Dialer-Specific Timed Call Not
                                                  Called
     789 - 792   4  # of Status Code #073 (1-9999) Dialer-Unknown Number
     793 - 796   4  # of Status Code #074 (1-9999) Dialer-Trunk Line Busy
     797 - 800   4  # of Status Code #075 (1-9999) Dialer-Not In Service
     801 - 804   4  # of Status Code #076 (1-9999) Dialer-Call Not Completed
     805 - 808   4  # of Status Code #077 (1-9999) Dialer-Rejected Record
     809 - 812   4  # of Status Code #078 (1-9999) Dialer-Modem/Fax Answered
     813 - 816   4  # of Status Code #079 (1-9999) Dialer-Disconnected
     817 - 820   4  # of Status Code #080 (1-9999) Dialer-Forced Resolved
     821 - 824   4  # of Status Code #081 (1-9999) Number Killed In Foneutil
     825 - 828   4  # of Status Code #082 (1-9999) Dialer-Number Changed
     829 - 832   4  # of Status Code #083 (1-9999) Bad Record
     833 - 836   4  # of Status Code #084 (1-9999) Dialer-Unknown
     837 - 840   4  # of Status Code #085 (1-9999) Timed Call With No Valid
                                                  Callback Time
     841 - 844   4  # of Status Code #086 (1-9999) Putfone With Record Number=0
     845 - 848   4  # of Status Code #087 (1-9999) Bad Stuff In Phone Record
     849 - 852   4  # of Status Code #088 (1-9999) Unknown Status
     849 - 852   4  # of Status Code #088 (1-9999) Unknown Status
     853 - 856   4  # of Status Code #089 (1-9999) Bad Special Value
     857 - 860   4  # of Status Code #090 (1-9999) No Stack Found
     861 - 864   4  # of Status Code #091 (1-9999) Duplicate Found In Fonebuld
     865 - 868   4  # of Status Code #092 (1-9999) No Next Bucket/Stack
     869 - 872   4  # of Status Code #093 (1-9999) Bad Phone Number
     873 - 876   4  # of Status Code #094 (1-9999) Maximum Number of Calles Made
     877 - 880   4  # of Status Code #095 (1-9999) Timed Number Too Old To Call
     881 - 884   4  # of Status Code #096 (1-9999) CfMC No Call List Drop In
                                                  Fonebuld
     885 - 888   4  # of Status Code #097 (1-9999)
     889 - 892   4  # of Status Code #098 (1-9999) All History Slots Used
     893 - 896   4  # of Status Code #099 (1-9999) Call Slots Used
     897 - 900   4  # of Status Code #100 (1-9999) Max Call Limit Met
     901 - 904   4  # of Status Code #101 (1-9999) No Answer
     905 - 908   4  # of Status Code #102 (1-9999) Busy
     909 - 912   4  # of Status Code #103 (1-9999) Two Busies (No Answer)
     913 - 916   4  # of Status Code #104 (1-9999) Scheduled Callback
     917 - 920   4  # of Status Code #105 (1-9999) Non-Scheduled Callback
     921 - 924   4  # of Status Code #106 (1-9999) Dialer-Nuisance/Abandoned
Call
```

```
925 - 928   4  # of Status Code #107 (1-9999) Dialer-Answering Machine
929 - 932   4  # of Status Code #108 (1-9999) User No Answer Statuses
933 - 936   4  # of Status Code #109 (1-9999)
937 - 940   4  # of Status Code #110 (1-9999)
941 - 944   4  # of Status Code #111 (1-9999)
945 - 948   4  # of Status Code #112 (1-9999)
949 - 952   4  # of Status Code #113 (1-9999)
953 - 956   4  # of Status Code #114 (1-9999)
957 - 960   4  # of Status Code #115 (1-9999)
961 - 964   4  # of Status Code #116 (1-9999)
965 - 968   4  # of Status Code #117 (1-9999)
969 - 972   4  # of Status Code #118 (1-9999)
973 - 976   4  # of Status Code #119 (1-9999)
977 - 980   4  # of Status Code #120 (1-9999)
981 - 984   4  # of Status Code #121 (1-9999)
985 - 988   4  # of Status Code #122 (1-9999)
989 - 992   4  # of Status Code #123 (1-9999)
993 - 996   4  # of Status Code #124 (1-9999)
997 - 1000  4  # of Status Code #125 (1-9999)
101 - 1004  4  # of Status Code #126 (1-9999)
1005 - 1008 4  # of Status Code #127 (1-9999)
1009 - 1012 4  # of Status Code #128 (1-9999)
1013 - 1016 4  # of Status Code #129 (1-9999)
1017 - 1020 4  # of Status Code #130 (1-9999)
1021 - 1024 4  # of Status Code #131 (1-9999)
1025 - 1028 4  # of Status Code #132 (1-9999)
1029 - 1032 4  # of Status Code #133 (1-9999)
1033 - 1036 4  # of Status Code #134 (1-9999)
1037 - 1040 4  # of Status Code #135 (1-9999)
1041 - 1044 4  # of Status Code #136 (1-9999)
1045 - 1048 4  # of Status Code #137 (1-9999)
1049 - 1052 4  # of Status Code #138 (1-9999)
1053 - 1056 4  # of Status Code #139 (1-9999)
1057 - 1060 4  # of Status Code #140 (1-9999)
1061 - 1064 4  # of Status Code #141 (1-9999)
1065 - 1068 4  # of Status Code #142 (1-9999)
1069 - 1072 4  # of Status Code #143 (1-9999)
1073 - 1076 4  # of Status Code #144 (1-9999)
1077 - 1080 4  # of Status Code #145 (1-9999)
1081 - 1084 4  # of Status Code #146 (1-9999)
1085 - 1088 4  # of Status Code #147 (1-9999)
1089 - 1092 4  # of Status Code #148 (1-9999)
1093 - 1096 4  # of Status Code #149 (1-9999)
1097 - 1100 4  # of Status Code #150 (1-9999)
1101 - 1104 4  # of Status Code #151 (1-9999)
1105 - 1108 4  # of Status Code #152 (1-9999)
```

```
1109 - 1112 4  # of Status Code #153 (1-9999)
1113 - 1116  4  # of Status Code #154 (1-9999)
1117 - 1120  4  # of Status Code #155 (1-9999)
1121 - 1124  4  # of Status Code #156 (1-9999)
1125 - 1128  4  # of Status Code #157 (1-9999) Trunk Line Busy
1129 - 1132  4  # of Status Code #158 (1-9999) Busy Status
1133 - 1136  4  # of Status Code #159 (1-9999) Busy Status
1137 - 1140  4  # of Status Code #160 (1-9999)
1141 - 1144  4  # of Status Code #161 (1-9999) User defined Scheduled
                                               Callback
1145 - 1148  4  # of Status Code #162 (1-9999)
1149 - 1152  4  # of Status Code #163 (1-9999)
1153 - 1156  4  # of Status Code #164 (1-9999)
1157 - 1160  4  # of Status Code #165 (1-9999)
1161 - 1164  4  # of Status Code #166 (1-9999)
1165 - 1168  4  # of Status Code #167 (1-9999)
1169 - 1172  4  # of Status Code #168 (1-9999)
1173 - 1176  4  # of Status Code #169 (1-9999)
1177 - 1180  4  # of Status Code #170 (1-9999)
1181 - 1184  4  # of Status Code #171 (1-9999)
1185 - 1188  4  # of Status Code #172 (1-9999)
1189 - 1192  4  # of Status Code #173 (1-9999)
1193 - 1196  4  # of Status Code #174 (1-9999)
1197 - 1200  4  # of Status Code #175 (1-9999)
1201 - 1204  4  # of Status Code #176 (1-9999)
1205 - 1208  4  # of Status Code #177 (1-9999)
1209 - 1212  4  # of Status Code #178 (1-9999)
1213 - 1216  4  # of Status Code #179 (1-9999)
1217 - 1220  4  # of Status Code #180 (1-9999) Dialer-Call Back Later
1221 - 1224  4  # of Status Code #181 (1-9999) Dialer-Got A Connect Than
                                               Abort
1225 - 1228  4  # of Status Code #182 (1-9999) Busy Changed To No Answer
1229 - 1232  4  # of Status Code #183 (1-9999) Dialer-No Ring Back
1233 - 1236  4  # of Status Code #184 (1-9999)
1237 - 1240  4  # of Status Code #185 (1-9999)
1241 - 1244  4  # of Status Code #186 (1-9999)
1245 - 1248  4  # of Status Code #187 (1-9999)
1249 - 1252  4  # of Status Code #188 (1-9999)
1253 - 1256  4  # of Status Code #189 (1-9999)
1257 - 1260  4  # of Status Code #190 (1-9999) Special Interviewer 1
1261 - 1264  4  # of Status Code #191 (1-9999)
1265 - 1268  4  # of Status Code #192 (1-9999)
1269 - 1272  4  # of Status Code #193 (1-9999)
1273 - 1276  4  # of Status Code #194 (1-9999)
1277 - 1280  4  # of Status Code #195 (1-9999)
1281 - 1284  4  # of Status Code #196 (1-9999)
```

```
1285 - 1288  4  # of Status Code #197 (1-9999)
1289 - 1292  4  # of Status Code #198 (1-9999)
1293 - 1296  4  # of Status Code #199 (1-9999)
1297 - 1300  4  # of Status Code #200 (1-9999)
1301 - 1304  4  # of Status Code #201 (1-9999) Special Interviewer 1
1305 - 1308  4  # of Status Code #202 (1-9999)
1309 - 1312  4  # of Status Code #203 (1-9999)
1313 - 1316  4  # of Status Code #204 (1-9999)
1317 - 1320  4  # of Status Code #205 (1-9999)
1321 - 1324  4  # of Status Code #206 (1-9999)
1325 - 1328  4  # of Status Code #207 (1-9999)
1329 - 1332  4  # of Status Code #208 (1-9999)
1333 - 1336  4  # of Status Code #209 (1-9999)
1337 - 1340  4  # of Status Code #210 (1-9999)
1341 - 1344  4  # of Status Code #211 (1-9999)
1345 - 1348  4  # of Status Code #212 (1-9999)
1349 - 1352  4  # of Status Code #213 (1-9999)
1353 - 1356  4  # of Status Code #214 (1-9999)
1357 - 1360  4  # of Status Code #215 (1-9999)
1361 - 1364  4  # of Status Code #216 (1-9999)
1365 - 1368  4  # of Status Code #217 (1-9999)
1369 - 1372  4  # of Status Code #218 (1-9999)
1373 - 1376  4  # of Status Code #219 (1-9999)
1377 - 1380  4  # of Status Code #220 (1-9999)
1381 - 1384  4  # of Status Code #221 (1-9999)
1385 - 1388  4  # of Status Code #222 (1-9999)
1389 - 1392  4  # of Status Code #223 (1-9999)
1393 - 1396  4  # of Status Code #224 (1-9999)
1397 - 1400  4  # of Status Code #225 (1-9999)
1401 - 1404  4  # of Status Code #226 (1-9999)
1405 - 1408  4  # of Status Code #227 (1-9999)
1409 - 1412  4  # of Status Code #228 (1-9999)
1413 - 1416  4  # of Status Code #229 (1-9999)
1417 - 1420  4  # of Status Code #230 (1-9999)
1421 - 1424  4  # of Status Code #231 (1-9999)
1425 - 1428  4  # of Status Code #232 (1-9999)
1429 - 1432  4  # of Status Code #233 (1-9999)
1433 - 1436  4  # of Status Code #234 (1-9999)
1437 - 1440  4  # of Status Code #235 (1-9999)
1441 - 1444  4  # of Status Code #236 (1-9999)
1445 - 1448  4  # of Status Code #237 (1-9999)
1449 - 1452  4  # of Status Code #238 (1-9999)
1453 - 1456  4  # of Status Code #239 (1-9999)
1457 - 1460  4  # of Status Code #240 (1-9999)
1461 - 1464  4  # of Status Code #241 (1-9999)
1465 - 1468  4  # of Status Code #242 (1-9999)
```

```
1469 - 1472  4  # of Status Code #243 (1-9999)
1473 - 1476  4  # of Status Code #244 (1-9999)
1477 - 1480  4  # of Status Code #245 (1-9999)
1481 - 1484  4  # of Status Code #246 (1-9999)
1485 - 1488  4  # of Status Code #247 (1-9999)
1489 - 1492  4  # of Status Code #248 (1-9999)
1493 - 1496  4  # of Status Code #249 (1-9999)
1497 - 1500  4  # of Status Code #250 (1-9999)

Other info:
1501 - 1580  80  Questionnaire filename
1581 - 1599  20  Blow Error Number (If interview "blows up")
1601 - 1632  32  Study code
1633 - 1652  20  Phone number being called

Interviewer message to supervisor:
1901 - 2000 100  Interviewer commands to supervisor via !spc,e,1

User area:
2001 - 3000 1000 User text area for !sys,I statement
```

4.4.4 Setting Up the Control Files Used by Survent

You will most likely only set most of these control file parameters once, but they greatly affect how your system operates. These files all reside in the directory CONTROL in the CfMC software area:

Filename	**Required?**	**Purpose**
FONESTAT	No	has labels displayed when showing call history statuses
INITIAL	No	has startup parameters read by all CfMC programs
MENTINIT	No	has startup commands affecting Mentor and the utilities
MSGFILE	Yes	file of messages and commands used in all programs
PARMFILE	Yes	file of parameters used by the CfMC server and Survent
SUPRINIT	No	has startup commands for supervisor program

TTYINFO	Yes	file of device descriptions used by the CfMC server
PHONEDEF	Yes	file of variable definitions used on SELECT statements

The FONESTAT file contains labels used by Survent and FONEUTIL when displaying call histories. You can modify the text to match the standard phone statuses you use on your system. There are 250 possible statuses that can have replacement text. Each line consists of a 3-digit number, a colon (:), and up to 28 characters of replacement text, for example:

```
EX:
101:NOT ANSWERED
102:ON PHONE
...etc.
```

The INITIAL file usually contains “meta” commands; you could for instance, set >DEFINEd variables that may be used by some programs. See the file INITEXAM in the CONTROL directory for examples of uses.

The MENTINIT file contains parameters used by Mentor and the Utilities. See the file MENTEXAM for examples of uses.

The MSGFILE contains numbered messages used by the program. You can edit the file MSGFILE.RAW and run it through the MAKEMSG program to make permanent changes to messages. If you are using multiple languages you can set the standard error messages for each language here. You may also use the command {!ERROR_MSG #### text} in PREPARE to override messages from the MSGFILE for a particular study. The MSGFILE is required by the system. See the Utilities manual for more information.

The PARMFILE contains most of the parameters used by Survent and the CfMC SERVER. A description of each parameter is provided below. The parameters are in the format “PARAMETER: value”:

```
EX:
TIMEZONE: 08
```

The PHONEDEF file contains definitions of all the standard CFMC variables available to be used by many commands used by FONEUTIL and SURVSUPR'S HIDE and REVEAL commands when entering descriptions for the SELECT statement. You may edit this file to include your own descriptions. *See 2.6 CONDITION STATEMENT* for possible syntax to use for the descriptions.

Here is an example:

```
EX:
>DEFINE @PHONE_NUM [1.10]
```

The SUPRINIT file contains commands used by the supervisor program at startup. It is stored in the CfMC Support directory. See the file "suprexam" in the support directory for examples, or see *4.4.2 SURVSUPR* above for a description of commands you might use.

The TTYINFO file contains an entry for devices that have standard device numbers in UNIX systems, or defined device numbers for DOS systems using a dialer. Each line defines the terminal type, device number, dialer extension#, etc. for the device. This is described in complete detail below. If you are running DOS without a dialer or setting LDEV descriptions using variables in UNIX this file may be blank.

USING THE TTYINFO FILE (UNIX)

The TTYINFO file contains device definitions for Survent. TTYINFO contains device definitions for "hard-wired" devices. You can provide the same definitions using variables for devices connected via telnet or some other networked access; in this case, you would assign the device information as part of the login script for the device (see below).

TTYINFO File Settings

The TTYINFO file contains one line for each ldev (logical device) you wish to define. It includes the ldev (or station) number, the ldev name (DOS/UNIX only), the type of terminal connected to the ldev, the time zone in which the ldev is located, the extension of the phone set co-located with the ldev (if a dialer is used), the sound channel (if a sound server is used), and the "socket" (if the device talks to the CFMC server using TCP/IP communications). The time zone, extension, sound channel, and socket are optional; if the time zone is missing it will be set to the time zone specified in PARMFILE.

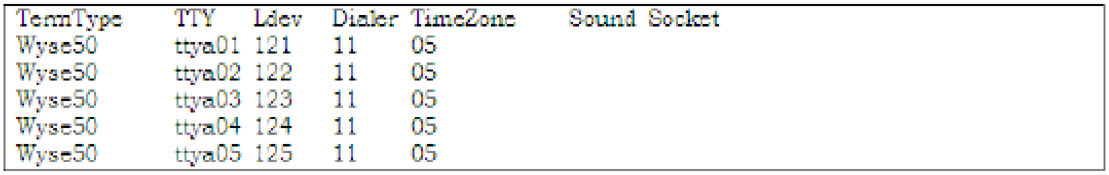
The TTYINFO file must be sorted by ldev name or a warning is generated.

The syntax for DOS or UNIX is:

```
termtype ldevname ldev# extension time zone soundchannel socket
```

If you are using a dialer, you must have valid information in your ttyinfo file for each station/extension you will be using on the dialer, even if you would otherwise use environment variables to describe your terminals.

Your ttyinfo file would look something like this:



```
TermType    TTY    Ldev  Dialer TimeZone    Sound Socket
Wyse50      ttya01 121   11     05
Wyse50      ttya02 122   11     05
Wyse50      ttya03 123   11     05
Wyse50      ttya04 124   11     05
Wyse50      ttya05 125   11     05
```

Notice the TermType and TTY values can be the same for every station, but the ldev/dialer correspondence must be correct.

Each line must have the indicated fields, separated by commas or blanks. Here is what each means:

terminal type is the terminal emulation you are using. The valid terminal types are: ADDS, ADM3, ANSI, ATT705, DIALUP, HP (HP2932 block mode), HP70041, HP70096. HP70098, IBM3151, UNKNOWN, VT100, VT220, WYSE50, WYSE85, WYSE150, and WYSE60

ldev name/tty is the name assigned to the device; this is arbitrary in DOS. ldev number is the number assigned to the device. It can be whatever number you choose. This is the number used by a supervisor when they start stations. May be 1-4000.

dialer extension is the phone's extension co-located with this ldev, if any; if connected with a dialer. Must have a value here if specifying timezone, sound channel, or socket. Must be within the range of numbers on the DIALER: parmfile command.

time zone is the time zone that the interviewer is in (1-24). The continental U.S. is time zones 5-8, going East to West, Paris is 23, London is 24. If left out, this defaults to the time zone specified in the PARMFILE.

sound channel is the sound channel used by this device, if any. This must be in the range of numbers on the SOUND_SERVER: command.

socket is the "socket" for TCP/IP interface with the CFMC server used by this device, if any. This is used by webSurvent and may speed up the server if used. This number must be greater than 2000 so as not to conflict with other sockets, and is usually LDEV+3000 for simplicity.

You can specify the values for these using environment variables instead of entering the lines in the TTYINFO file for stations that are not Hardwired to the computer (eg. TELNET stations). The environment variables and their uses are as follows:

UNIX name	Definition
LDEV	device number for the station (1-9999)
TERM	terminal type, for example, “wyse50”
EXTENSION	phone extension # for the dialer
CFMCPROCESSTIMEZONE	timezone (if different from the server)
SOUNDCHANNEL	sound channel for the sound server

SOCKET	socket number if using sockets (need SOCKETS: YES in the PARMFILE)

To set values for these variables in Unix, you would put the "setenv <VARIABLE> <value>" commands in the login script that is used for a particular device, e.g., "setenv LDEV 123" for station 123. Variable names must be UPPER case.

The interviewer can also override the terminal type specified in the TTYINFO at the interviewer ID prompt by specifying T=name or T=## (see *4.2 INTERVIEWING WITH SURVENT)*. Here is the list of support terminal types:

Terminal type	
Alphanumeric	**Numeric**
ADDS	11
ADM3	10
ANSI	1
ANSI25	18
ATT705	13
AT385	20
DIALUP	2
HP	9
HP70041	12
HP70096	14
HP70098	15
HPTELNET	21
IBM3151	8
LINUX	22
SCOANSI	19
VT100	17

VT220	4
WY50	5
WY60	16
WY85	7
WY150	6
UNKNOWN	3
XTERM	23

If the time zone an interviewer is in is different from the time zone the server is running in, and this is reflected in the time zone setting for that interviewers device, it will correct the 'TIME HERE' time echoed when scheduling callbacks for the nterviewer's time zone.

Setting Parameters in the CfMC PARMFILE

The PARMFILE must reside in the CFMC CONTROL directory, unless the CFMCPARMFILE variable is set to allow it to point somewhere else. DOS and UNIX require one PARMFILE for every server that is run (in the environment's CONTROL directory by default).

Required PARMFILE Options

There are three required parameters in the PARMFILE.

1 **EXPIRATION:** This tells the CfMC server how many stations may be running concurrently and other pertinent information.

2 **VALIDATION:** This is a 14-character string that is an encoded version of the expiration string. It is case-sensitive.

Both strings must be updated at release times (even if you are not updating the version of the software you are using) and at times that you have increased the number of stations.

3 **TIME_ZONE: ##** This tells the CfMC server the time zone the computer is in for purposes of call scheduling across time zones. Time zones are numbered 01-24 relative to Greenwich, England (GMT). "TIMEZONE: 08" is would be used if you were in the Pacific time zone in the United States.

Optional PARMFILE Options

Below are the nondialer-related parameters in the PARMFILE:

<u>Interviewer Related Options</u>

AFTER_BLOW: BLOW/SUSPEND/CONTINUE controls what to do when a blow error occurs.Blow errors are errors in questionnaire design or CfMC system errors that cause the interview to terminate abnormally.

The default is to BLOW (that is, stop the interviewer session, save a data record, and print an error message). The SUSPEND option tells Survent to make a suspend file from the question asked last, Then stop the interviewing session. The CONTINUE option makes the suspend file.

Then interviewing is allowed to continue. This allows interviewers to resume suspended records once the cause of the blow has been found and fixed. Use CONTINUE option with care because interviewers may continue interviewing and ignore the problem, without reporting it.

BACKUP_CMD: allows you to have a string for interviewers to use to back up in the interview in addition to the standard "^" command. BACKUP_CMD: BBB would allow interviewers to type "BBB" to back up to the previous question instead of "^" at a command prompt. This will now allow you to use BBB on a Highlightcats screen though (See Highlightcats mode).

However, the special character "<" is designed to operate like "^" on Highlightcats Mode questions if "BACKUP_CMD: <" is set. This is because the "^" is difficult to type on some non-U.S. keyboards and "<" is not.

BACKWARDS_SKIPS=### controls how many backwards skips are allowed. This keyword controls the processing of backwards goto or skipto statements. By default the number is 500. This is to keep interviews from getting into a "loop" situation where they go backwards over and over. When the maximum number of backward skips is reached, the questionnaire will get a blow error. We have the keyword because some applications want to allow more backward skips.

DATE_FORMAT:DDMMYY will allow the interviewers to enter callback times with the day first, rather than the month having to be first. This is the format used in Europe. There is an additional parameter to control tdefault is upper case but you can say: DATE_FORMAT: MONTH_NAME_LOWER_CASE

To change it to lower case. The other keywords are "MONTH_NAME_UPPER_CASE" and "MONTH_NAME_FIRST_UPPER" (first letter upper case, rest lower case). You can combine these keywords with the "DDMMYY" or "MMDDYY" parameters to control month and day display order.

DONTREDOSUSPENDBLOCK:YES causes the program not to re-execute the SUSPEND block when interviews are resumed. By default it does redo the suspend block.

DUPLICATE_INTERVIEWER_IDS:NO disallows interviewers with the same interviewer ID from logging in at the same time. This was added to avoid having data with the wrong interviewer ID assigned to the complete. If this is on and someone enters a duplicate interviewer ID, they will see:

```
Interviewer "xxxx" already logged on, duplicate
logins not allowed!
```

Then, the program will reprompt for a new interviewer ID. Interviewers in PRACTICE mode or DEBUG mode are exempt from this check. They can log in with the same ID as anyone else. The default is still that duplicate interviewer IDs are OK.

FAILEDRESUME: RESTART/STARTHERE/BLOW allows you to control what will happen to interviews when a resumed interview fails to match the current interview (in cases where you've made changes to your questionnaire). The options are:

Option	Definition
RESTART	Causes the interview to be restarted from the beginning if the suspend doesn't match the current questionnaire version (default).

STARTHERE	attempts to restart interviews where they were suspended if the questionnaire has been modified. If there is not a match, it starts just prior to the place the old version stops matching the current version. This keeps the respondent from having to answer Questions again.
BLOW	causes the interview to BLOW instead of being resumed. If the suspend file does not match the questionnaire it is resuming under. This allows you to save the data from the original interview without continuing a changed interview that may not make any sense.

This option may be overridden on a study level by using the FAILED_RESUME study header option in the spec file.

INTERVIEWER_LOGGING: YES/NO/AFTER_EVERY_QUESTION This parameter sets defaults for logging. The default is "Yes". Previously, this had to be stated in the header of all questionnaires you wanted to log or specified using supervisor "log" and "stoplog" commands.

Interviewer logging saves all the interviewer keystrokes and answers to question in the file called log<intv id> in the CfMC ipcfiles directory (<intr id> is the interviewer's interviewer ID). New interviews are appended to the file across studies. The options are:

```
INTERVIEWER_LOGGING: Yes - Turns on logging
INTERVIEWER_LOGGING: No - Turns off logging
INTERVIEWER_LOGGING: After_every_question - logs
after every question
```

Be careful when using the "after_every_question" option because this may cause a heavy disk Load on the system. You can also set and unset these on a study basis using the same-named header option, or override them by using the "log" and "stoplog" commands in the supervisor.

SHOW_STUDIES:YES This causes Survent to list the available questionnaires and their study comments on the screen when starting NETSURV or when using the CHI command to change toanother interview. The keyword is specified in the CfMC Parmfile as follows:

```
EX:
SHOW_STUDIES: YES
```

When you get the screen, you can press ESC and enter the name of the questionnaire you want (as usual), or you can scroll to the study you want and press <ESC> to choose that study.

The program knows which questionnaires are compatible with the version of Survent you are running.

TEXT_START_INSERT_MODE: YES sets the mode when entering TEXT question data to be INSERT mode instead of the default of OVERSTRIKE mode (UNIX/DOS only). This more closely matches other editors.

Sample File Related Options

ASCIIPHONERECTYPE: RAW/SQUISH allows you to control the size of converted ASCII phone files created by FONEUTIL and read by FONEBULD. The default is “RAW”, the file is written to the full size of the record (about 10000 bytes per record). If you specify “SQUISH” the file will be written in a compressed format, where it places a letter in column 21 from A-Z as the number of 200 byte blocks of text the file has, and writes the CfMC variables and call histories after the user text. This can make the file as small as 1,000 bytes per record.

DO_NOT_CONTACT_FILE: <filename> The "do not contact" file is a file of numbers that is checked against whenever you add numbers to a sample file. It is an indexed CfMC TR file with phone numbers assigned as the case id for the file. To set this up so that you always check against the file, in the CfMC PARMFILE (located in the CONTROL directory) add the line:

```
      EX:
DO_NOT_CONTACT_FILE: /usr/cfmc/control/badrecs.tr (UNIX)
DO_NOT_CONTACT_FILE: f:\cfmc\control\badrecs.tr (DOS)
```

The file (eg. badrecs.tr) can be located anywhere you like as long as you have read access to the file. If a telephone number is present in the do not contact file, the number will not be called.

To create or append to a DNC file, use MENTOR with the following commands:

```
EX:
~input <ascii filename> ascii=100 id=1.10
~output <dnc filename> trfiledirectory=<####>
maybecreate writenow
~end
```

Where "####" is the maximum number of records you expect in the file.

Survent's command to add a number to the phone file (!PHONE,A) checks the file before adding the number, in addition to FONEBULD. Any number which can not be called is added to the fonefile in stack 330, the 'never-call' stack, and given resolved status 96.

DO_NOT_CONTACT_PREFIX_FILE: <filename> checks the first 7 digits of a phone number, and if it matches a record in the file specified, it rejects the number instead of adding it to the phone sample file. This is designed to disallow calls to cell phone blocks, but can be used to disallow calls to any prefix you want.

To build the file, write something like:

```
EX:
~input <ascii filename> ascii=100 id=1.7
~output <dncprefix filename> trfiledirectory=<####>
maybecreate writenow
~end
```

where <####> is the number of phone number prefixes you expect to have. To invoke the file, in fonebuld say "DO_NOT_CONTACT_PREFIX_FILE= xxxx" where xxxx is the name of the .tr file to use (the .tr extension is not required in the command). To set a system default, add the command "DO_NOT_CONTACT_PREFIX_FILE: xxxx" in the CfMC parmfile. You can also shorten the command to "DNCPREFIXFILE".

DUPLICATE_FONE_CHOOSER: NEW/OLD This command was added for clients who want back compatibility to the previous version of the duplicate phone number screen. The default is "NEW", to use the old one use "OLD". The new screen has many enhancements, including allowing more numbers, more text, and an interactive display, but the old one worked fine in most cases and has a different look and feel so we are letting clients choose.

END_OF_DAY:HH:MM starts a new log file so that each day's logs can be combined. The END_OF_DAY parameter tells the system when to make a new log so it is clear when the start of the next day's log is.

```
EX:
END_OF_DAY: 02:00
```

The example asks that at 2 a.m., build a new server log file (and phone record log file if Server_write_ascii_phone_records:Yes), and if you go to get a timed call at that time, integrate the phone file for that study to set up time calls for the next day. Therefore, all the server log files named with date times after the END_OF_DAY time and before the END_OF_DAY time will be included in reports for that day.

This command also controls when to integrate phone files automatically to set calls into the "today" calling stacks. If END_OF_DAY is 2:00 a.m., the phone file will not be integrated until the first time it tries to get a number after 2:00 a.m. that day.

FONE_HISTORY_START=### is used by clients who use nonstandard formats for the phone call history area. This allows them to specify valid locations in their questionnaire specs when referencing phone history variables.

FORBIDDENFILE: <filename> tells the program the name of the file to be used by FONEBULD when determining whether phone records may be used. Any phone numbers listed in the "forbidden" file will be marked as "unusable" in the phone file so you don't mistakenly call numbers of people who do not wish to be called, etc. The FORBIDDENFILE <filename> command in

FONEBULD causes it to look to that file instead of this one for forbidden records.

For more details on FORBIDDENFILE, *see Chapter 6, section 6.1.1 Building the Phone File,* Phone System Commands.

FORCE_INTEGRATE: #### will not allow the study to start if your phone file exceeds the specified # of timed callbacks. It will force users to "integrate" the phone file using FONEUTIL offline before the study can be started. The reason for this is so that the server is not overloaded rescheduling the numbers when it is busy with other studies.

The default # is 200, and the number can be from 1-9999. Use caution when setting this number higher because interviewing may be slowed down significantly until the server processes all the numbers and interviewing may be live on other studies.

HARD_BUCKET_SCHEDULING:YES this disallows the scheduling of calls beyond the times specified as bucket times 1 (start of day) and 4 (end of day). By default, timed numbers may be scheduled as long as they are within SHOPOPEN and SHOPSHUT times for the day.

HARD_BUCKET_TIME:YES will cause any phone number which comes up outside the specified limits of bucket times 1 and 4 to be put back to be called another time. Timed calls will be scheduled for tomorrow at the same time. Owned or special records will be put back as they were, at the end of the stack from which they came.

MARKETWEIGHT_ZERO_STATUS: ### gives the # status automatically to calls to numbers in a market which had a weight of 0. This will apply to timed calls and numbers that were in thespecial interviewer stacks. Because some statuses are used by CfMC and some statuses cause inappropriate actions, the marketweight_zero_status command has been limited to the following statuses:

- Release number to interviewer 0
- Resolved statuses 10-70
- Non-specific callback statuses 107-156 or 191-199

- Non-specific callback statuses/No history 201-209 or 213-214

SERVER_WRITE_ASCII_PHONE_RECORDS: YES This command saves a copy of the ASCII phone record each time you write a status to the phone file. The purpose is so that you can get real-time phone reports or transfer the data to other systems without having to convert records from your phone files.

The server writes to a file LP<DDHHMM> (DOS) or lp<DDHHMM.YYYY_MM> (UNIX) in whatever directory the server is running from (###### is the day, hour, and minute the file was created, YYYY is the year and MM is the month).

If you type "server:log" at a supervisor, the server closes the current lp###### file and starts a new one, so you would want to do this just before running your application.

NOTE: The "END_OF_DAY: HH:MM" parameter will cause the program to start a new log file so that each "day" has all the records for that day.

SPECIAL_ONLY_SPECIAL: YES this causes special interviewer types to NOT get any numbers that are not marked as “special” numbers. The default is that once there are no special numbers available, they start getting the regular numbers until more special numbers are available.

UPDATEFONEHEAD: YES will update the phone file header as soon as someone uses the MPF command in SURVSUPR. This will make changes immediately available in Survent. Otherwise, updates are done about once a minute.

<u>Sample File-Related Options/Foneutil commands</u>

SURVSUPR now supports most FONEUTIL commands while live.Most of the commands that previously could only be run offline while a study is down may now be done in the supervisor while the study is live. These commands also are incorporated into the webSuper Utilities program. (See the *webSuper Manual* for more on these commands).

The commands supported are:

PHONE_LIST <study><select>: List sample to screen or to LPDEV if specified.

PHONE_ZAP <study> <Last Save> <select>: Remove call histories and make sample "new". “Last” removes last call history only. “Save” removes no histories but changes stack.

PHONE_ERASE <study> <select>: Same as Zap but only on active sample.

PHONE_KILL <study> <select>: Resolve sample with status 81 immediately.

PHONE_ALTER_TIMED <study> <select> <time>: Changes time of callbacks.

PHONE_RETURN_OWNED_NUMBERS <study> <select>: Returns numbers from "owned" stack to original stack.

PHONE_GATHER_SPECIALS <study> <select>:Move numbers from other stacks to "special interviewer" stacks.

PHONE_SORT_SPECIALS <study> <1-9>: Sort "special interviewer" stacks by time zone for proper call order. Use “1” for instance to sort special interviewer type 1 stack ***only.***

Supervisor Options

ANYONECANMODIFY: YES/NO controls whether others may use the SURVSUPR MPF command. This parameter overrides "ownership" of a phone file by a particular supervisor when you have multiple supervisors on your system. Setting this to YES allows any supervisor to use the MPF command at the same time on the same study.

WARNING: Use of this command may cause a supervisor's changes to be overwritten by another supervisor if they issue the commands at the same time.

DASHBOARD:NO keeps SURVSUPR’s DAI, DIS, QSS, and SPI screens from updating themselves automatically.

NOSTARTRANGES: YES disallows RANGES of interviewers to be started by the supervisor. This means you cannot specify “STS 1-100 BANK”, you may only specify “STS 1,2,3,4,5,etc. BANK”. It keeps supervisors from making mistakes and starting stations that were not intended to be started.

SUPERPASSWORDS: MPF=<password> lets you assign a password to modify the phone header for all jobs. Can also specify passwords for AT, MID, MPF, QSS, QSS MODIFY, MARKET, NQS and SERVER: commands by listing them consecutively with their password on the command line.

SUPERPASSWORDS: ACTIVATE/DEACTIVATE. You can now put passwords on SURVSUPR "activate" and "deactivate" commands in the SUPERPASSWORDS: list to secure the commands. For example:

```
SUPERPASSWORDS: ACTIVATE=GOAHEAD DEACTIVATE=AREYOUSURE
```

SUPER_REPROMPT_ON_COMMAND: YES/NO determines whether you are returned to "RUN" mode after entering a command or kept at the command prompt for another # seconds (controlled by the "PAUSE ##" keyword. If you wish to return to "run" mode, then use the "no" option by adding the command to the parmfile.

Dialer-Related Options

CALL_INCOMPLETE_STATUS: ### says what the status returned from the dialer for numbers with an "incomplete" status will be; the default is 76, but you may set it to a value from 2-250.

The main reason for this feature is to allow you to call these numbers back by giving them a callback status (such as 132) instead of a resolved status.

CALL_MODEM_STATUS: ### says what the status returned from the dialer for numbers returned with a "modem" status will be. The default for modem status is 78. The idea behind this feature is to allow you to call these numbers back by giving them a callback status (such as 132) instead of a resolved status.

DEFAULT_USE_DIALER: YES causes any non-practice mode supervisor->interviewer startups on the system to use a dialer, unless otherwise specified. This means all supervisor "Start" and "Chi" commands will use the dialer unless "-Dialer" is specified on the command.

DIALER##: If you are using a dialer, there must be one line in the PARMFILE for each dialer (up to 20 dialers under one server are supported). The syntax is:

```
DIALER##:<ip address><read port><write
port><extensions>[SOUND]
[<baudrate>][<name>][<type>]

EX:
DIALER: 192.168.1.1 5001 5002 1-20+32+45+51-99
```

This says the dialer is at ip address 192.168.1.1, the read port is 5001, the write port is 5002, and the extension used are #1-20,32,45, and 51-99.

DIALER_CONNECT_STATUS: ## says the status to give to numbers returned connected by the dialer; the default is "1" but this can cause problems because the default for a "COMPLETE" is also "1" and if you don't otherwise set a status for numbers returned from the dialer they will always get marked as if they were a "complete".

DIALER_NUISANCE_STATUS: ### says the status to give numbers returned from the dialer with a "nuisance" status (numbers where the dialer hung up on the respondent because no interviewer was available). The default is "106" which is a callback status. You could set this to a resolved status like 75 if you don't want these numbers called again.

EIS_PREFIX: ### this causes numbers sent to the dialer to be sent with a prefix in front of the number. This is to handle cases where phone numbers need a "9" or such to dial out, or have a prefix to control which trunk line to use, etc.

EIS_REAL_EXTENSIONS: says to assign the same extension number in the EIS dialer GATEWAY screens as those assigned by the CfMC server when there are multiple dialers. By default, EIS assigns station numbers starting at "1" for each dialer.

FONERECNUMLEN: ###### says what length the phone record number sent to the EIS dialer is; the default is 5 digits with a max of 99,999 records. If set to 6, it will allow phone record numbers up to 999,999 when interacting with an EIS dialer. Note that the EIS dialer must be configured for 6-digit phone record numbers also.

MAX_DIALER_CLOCK_ERROR: <number of seconds> indicates when to disallow dialing because the CfMC server and dialer clocks don't match. It will require that the dialer clock and the server clock agree to within the specified number of seconds or the dialer will not run.

REMOTE signals the use of "VOIP" (Voice over IP) for the SER dialer. If you add ",remote" to the extension list on the DIALER##: line in the parmfile, the software will send the commands needed to connect to "VOICE OVER IP" phones. Here is an example:

```
DIALER01: 123.212.254.012,5000,5001,20001-20200,remote
```

NOTE: In order for a supervisor to be able to voice-monitor these stations, they have to log in with an extension that is greater than 20000.

SER_INTERVIEWER_NAME: NAME/ID/IDNAME/LDEVNUM/EXTENSION this command controls what to send as an interviewer name to the dialer. The default is the first 9 characters of the interviewer name field from the employee file (columns 5-13). The options are:

Option	Definition
NAME	first 9 characters of the name field (5-13) (default).
ID	just the 4 character interviewer ID (1-4)
IDName	Id from 1-4, followed by a dash, followed by first part of the name (5-9).
Ldevnum	The device number the interviewer is running at
Extension	Extension The phone extension number (old default in version 7.2)

Server-related/Miscellaneous Options

CASES_PER_DOT: Controls the display of dots when reading files. Mentor, Fonebuld and Foneutil programs display dots while reading files. The default is that one dot is displayed per 10 cases using Mentor and one dot per 100 cases using the FONE utilities. This can now be controlled by using the "CASES_PER_DOT: ###" parameter in the PARMFILE.

IGNOREQUOTACRC:YES causes Survent to ignore the fact that the quota file version may notmatch the questionnaire version. This makes it easier to manage studies where the questionnaire file or quota file is being changed often, but may cause errors in quotas if files don't match for number of or order of quotas. By default the program checks to make sure the quota names and order are the same in the QFF and QUO files.

LL_LOGGING: YES ##### LL/NO Turns on server logging of interviewing events (see 4.4.3, *SERVER LOGGING* above); Set to **NO** to turn off. If set to **YES**, you can also control the size of the log files and how they are named. The file size defaults to 20,000 lines but can be set to any value. The default file name is "LxDDHHMM", where "x" is a letter from A-L representative of months 1-12. If you say "LL" at the end of the parameter line, the filenames will always be "LLDDHHMM". Here is an example line:

```
LL_LOGGING: YES 10000 LL
```

This tells the program to perform logging, start a new log file each 10,000 lines, and name the files starting with "LL" regardless of the month.

LLONLY_LOGGING: YES This will cause the server to write log records LL01-LL16 to a separate file from the regular LL file where logging messages are written. This benefits users who use the LL records to run the CfMC log reports, because the size of the file is greatly reduced by excluding the other messages, which increases the speed of processing. The filename will have this format: LX<ddhhmm>.<yyyy>_<mm>. It will match the name of the LL file with the generic logging messages.

LOGFILETIME: # specifies a time in minutes for the server to make a new logfile. "LOGFILETIME: 2" will cause the server to close the logfile and open a new one every 2 minutes.

MINIMUM_BOOTH_IDLE_TIME: <number of seconds> controls how long you must wait before you can restart a station after it has been disconnected. This keeps the server from getting overloaded when the system attempts to stop and restart too many interviewers at the same time. The default is "0" seconds. So, this will only go into effect if you add the option.

REV_COUNTS_ON_BREAK_LUNCH: YES This determines whether the timers are reset when interviewers or supervisors type “lunch” or “break. If set to YES, the timers and number of completes and sessions will be reset to 0. The default is that they are NOT reset.

SAVE_OLD_SUSPEND_FILES: YES Survent saves RESUME files until the next interview is completed or saved with a “non-resume” status. This is so you can recover the suspend file in case there is a problem resuming. In addition, this parmfile command saves resumed files for posterity.

Files are saved in the subdirectory of the standard CfMCRESUME directory “<study>.r_/old” when interviews are completed or aborted.

SERVERDROPSTUDY: ### controls when the server may drop a study if no work is done. The setting (#) is in minutes, and will allow the server to close a study if there has been no activity on the study for the number of minutes specified. The default is not to drop studies at all, which leaves files open that you might otherwise want to work with. SERVERDROPSTUDY: 60 will close a study after 60 minutes of inactivity.

SERVER_MUST_BE_OWNER: YES (UNIX only) tells the server not to load the study unless it has the ability to set the file access rights for the study files. This prevents users from modifying or deleting files while studies are live.

SERVER_TIME_CHECK: <error/warn/reset> checks or sets remote device times to match CfMC server time. This command checks remote processes talking to the CfMC server's time to match the server's time. In particular, DOS interviewing stations running on Windows machines will get their machine times checked. This also checks other remote servers like Winsound.

The options are "error", "warn", and "reset".

The "error" option will display:

```
(error #4488) Local time differs from stdysrvr time by 360
              seconds
              Time difference is > 300 seconds, must stop now.
```

The "warn" option displays:

```
(warn #9999)  Local time differs from stdysrvr time by 360
               seconds
               May cause problems on reports with date/time
              information in them
```

The "reset" option displays:

```
(warn #9999) Local time differs from stdysrvr time by 360
              seconds
              Local clock reset for this run to match the
              stdysrvr
```

Any time a difference of greater than 300 seconds is detected, one of the above pairs of lines will be generated. The default is error.

SERVER_WRITE_INCREMENTAL_DATA_RECORDS: YES/ARCHIVE will back up all data as the records are written. The data is written to a file of the same name as the main datafile, but in the "current" sub-directory under the CFMCDATA directory. When opening a study, if necessary, the server will create directories 'current' and 'last' under the $CFMCDATA directory if it exists, otherwise under the directory the CfMC server is running in. Cases written to the <study>.tr file will be written to current/<study>.tr as well. Also, the supervisor can issue the command "sav(e)_inc(remental)_data <study>" and the CfMC server will close the current/<study>.tr and move it to last/<study>.tr and a new current/<study>.tr will be started. In this way, you can have "partial" datasets to do daily reports for instance. If the parmfile option is: server_write_incremental_data_records: archive, the CfMC server will do the above as well as taking the current/<study>.tr and append it to the file archive/<study>_YYYYMMDDHHMMSS.tr. Using this feature:

- The sum of all <study>.tr files will be a complete match for the official <study>.tr file.
- The most recent one is a match for the one in the "last" directory
- If phonesaysdatarec is used and there are multiple writes to the file, this will save each as a separate record.

SOCKETS: YES (UNIX only) the CfMC communications package now supports socket messaging if you have TCP/IP installed on your machine. This increases communications speed by a factor as high as 20; for example, the time between Web pages using webSurvent has decreased from 5-7 seconds to .2 seconds. It allows more data traffic on the system without potential system failures due to overload. This is independent of whether of not you use sockets for dialer or sound server interfaces, it only applies to the communication between the CfMC server and the stations.

If using this parameter, you must add an entry for each device in the TTYINFO to tell the program its socket number, for example:

```
wyse50 25 5 15 10 4001
```

This says the terminal type is wyse50, the CfMC device number is 25, the time zone for this station is 5, the extension for the dialer is 15, the sound channel is 10, and the socket is 4001. You can use any numbers for the socket number as long as they are exclusive.

A downside to using sockets is that if communications to the socket are lost to a station, it takes a minute for the socket to become available again.

SOUNDSERVER#(a): #(b) #(c)-#(d) (a) gives information about soundserver device(s), the machine number(s); (b) identifies which serial port to talk through, and (c-d) are the phone extensions that are being used.

SURVMON_SECURITY: <level> lets you automatically force SURVMON to use a certain access type. <level> is any of the following:

- ONESTUDY: You must specify which study to watch on the command line using the "study=" parameter, and may only watch interviewers on that study.
- ANYSTUDY: Watch any interviewers on a particular study at a time. You can also change studies, but while you are assigned to a particular study you cannot watch an interviewer on another study.
- ANYLDEV: Watch any interviewer at any time no matter what study you are assigned to.

SV_LOGGING: <FILENAME>,<LENGTH> tells the program the name and record length of the file for the server to save log messages to generated with the SPC,R statement in Survent.

TEST_ONLY: YES this allows you to compile questionnaires in DOS even if a security key is not installed on the machine. It will allow you to compile but will not allow you to collect data with SURVENT, so it is designed for testing questionnaires at home or sending test questionnaires to clients.

TTYINFO: NO causes the server to NOT look in the ttyinfo for device information about interviewer stations; the information would come from the setting of environment variables in this case (**UNIX**).

WEBSURVENT_STARTUP_APPEND allows you to specify parameters for the server to include when starting Survent sessions for Web interviews. For instance, "WEBSURVENT_STARTUP_APPEND: core:3000000" will allow the survents to start with 3 megabytes of memory instead of the default of 600,000 bytes. You can have any command line parameter you want, such as "define:" to set a variable or "init:" to send responses to the initial prompts. ***NOTE:*** The list of possible parameters are noted in the Utilities manual. You can also do this by setting the environment variable CFMCHTMLSURVAPPEND before starting the CfMC study server.

4.5 INTERVIEWS BY MAIL (DOS)

Disk-by-Mail

REQUIRED FILES:

SURVENT.EXE	Survent program file
SURVENT.CFG	Survent configuration file, which can be preset
FIELDMSG	program message file; must be in \CFMC\CONTROL and be renamed to MSGFILE. Can also use the standard MSGFILE, but it is larger than FIELDMSG
studyname.QFF	your questionnaire file
studyname.QUO	your quota file
PARMFILE	parameters file; must be in \CFMC\CONTROL. See previous pages.
DOS4GW.EXE	compiler file, goes wherever SURVENT.EXE goes.

Depending on the size of your QFF file, all of the required files should fit on a low-density diskette with space to store the data file.

When testing your disk, make sure your test environment reflects the environment of a normal respondent (i.e., your path doesn't point to the normal GO directory and the CfMC variable is not defined).

See *Appendix G: HARDWARE AND SYSTEM CHECKLIST* for necessary hardware configurations.

HELPFUL SURVENT FEATURES FOR SELF-ADMINISTERED QUESTIONNAIRES

Data Entry Controls:

- {!HIGHLIGHTCATS} allows the respondent to choose items from CAT or FLD response lists by either arrowing to an item or typing its response code. No data entry prompt is displayed.

See *3.2.1 INTERVIEWING CONTROL COMMANDS* for more information.

- Positioning and defining your own data entry prompt, _. (see *2.5.1 SCREEN FORMAT CONTROLS, Positioning The Data Entry Prompt*)
- {!AUTORETURN} Causes responses to be accepted as soon as they reach their maximum length. (See *3.2.1 INTERVIEWING CONTROL COMMANDS* for more information.)
- Rating scales, !NUM,R. This NUMERIC question subtype displays a rating scale centered on zero, allowing a response +/- the maximum value specified by using arrow keys to move to a position on the line. (See *2.4.5 NUMERIC QUESTION TYPE, Subtype R.*)
- · GRID questions let the respondent move around on the screen and answer question (possibly) out of order (See *3.1.2 SCREEN CONTROL STATEMENTS, GRID Question Block*).

Screen Format Controls (see 2.5.1 SCREEN FORMAT CONTROLS)

- SCREEN_LINES specifies the screen size to make sure your questions will fit on the interviewing device. (see *2.3.2 STUDY HEADER STATEMENT*)
- Using boxes to specify screen position, \(#,#,#,#)
- Using boxes to display a border around question text and/or response lists (\O).
- Use available screen to put questions in whatever screen space is available below the previous question (\A).
- Horizontal response list wrapping (\H).
- Text enhancements: bold, flashing, inverse video, underlined (\B, \F, \I, \U).
- Color enhancements (\C).

Interview Control

- ONE_INTERVIEW, causes Survent to exit after an interview is completed. (see *2.3.2 STUDY HEADER STATEMENT*)

- Displaying prior responses to either single or multiple response questions (see *2.5.3 DISPLAYING PRIOR RESPONSES AND DATA*)

Controlling Data File Layout

- CARD_ID_FORMAT=col.wid. Puts the case ID on every 80-column record in the data file and a record ID in the location specified here. (See *2.3.2 STUDY HEADER STATEMENT.*)
- MAXIMUM_LABELS=# controls the size of the compiled questionnaire file (QFF). (See *2.3.2* STUDY HEADER STATEMENT.)
- REFORMAT spec file compiler commands to reformat data files from compressed column binary to ASCII format, expanding CAT questions by spreading multi-punches, and listing responses to TEX questions. (See *5.7, REFORMAT.*)

4.6 SURVENT LOGGING

By default, the CfMC server logs all the responses an interviewer types, including all movement back and forth in a questionnaire, changes, and saving information after terminates or aborts.

One file is saved per interviewer across sessions. The file name is LOG<intv ID>. If you are running under the server, the file is saved in the CFG (UNIX) or IPCFILES directory/group (DOS). If running standalone Survent, the file is saved locally.

If logging is turned OFF for your site, there are three ways to cause logging to occur for a particular interviewer or study:

1. Add the option LOGGING to the questionnaire header statement, and every interviewer on that questionnaire will have their own log file created. LOGGING=AFTER_EVERY_QUESTION will set logging to Debug mode and will save the log file after each question instead of each interview.
2. In SURVSUPR, start interviewers using the "LOG" command instead of the "STS" or "START" command. Enter: LOG #<station list><study>.
3. The interviewer can enter LOG (to do regular logging) or LOGDEBUG (to save the log file after each question) at the **Return** to Interview--> prompt to log this session.

To turn off logging, enter -LOG at the **Return** to Interview--> prompt or use the STS command in SURVSUPR to restart interviewers without logging.

Here is an example of the log file:

```
''<Header: Bank Mike 03/07/95 3:00pm>
1 '':HAVECARD:    0.10
2,9 '':CARDTYPE:  0.20
American excess card '':OTHRCARD: 0.3
^ ''<control>
American Express card ''OTHRCARD: 0.3
abort ''<control>
''<Trailer: Bank Mike 03/07/95 0001 >
...
```

Only one log file is maintained per interviewer; new sessions append to the log file if it exists.

NOTE: Log files only store typed responses, not data recorded internally by your questionnaire.

4.6.1 Playbackfiles

Playback files record the events that occur in survent during an interview. What makes them different from any other log or list files is that they may be used as input to survent in order to reproduce the interviews that they recorded. A second unique feature is that they allow you to save what was presented to the interviewer in a separate file so that you can examine it later. This is particularly useful for web mode interviews. Two meta commands are associated with playback. Here they are with the options discussed below and their default values.

```
>playbackfiles

STORE_FILES_IN_STUDY_DIRECTORY
BASE_FILENAME= (uses study by default)
-MAKE_PRESENTATION_FILE
-ECHO_TELLS
-ADD_COMMENTS
APPEND
-SAME_LISTFILE
PLAYBACK_MODE=full_replay/answers_only (no default value)

>echo_presentation=

HTML (default)
SCREEN
LINE_MODE
```

The simplest use of this feature is to put >playbackfiles at the start of your survent session, although it's most commonly used in an initial file. (Remember you can have an initial file in ${CFMC}control, $HOME (on Unix), and your current working directory.) If your questionnaire looked like:

```
~prepare compile
[test1]

{Q1:
\@ Title text
!fld
1 Yes
2 No
}

~end
```

And you ran stand alone "survent con con" on this, answering "yes" to Q1 you would find that a directory named test1.p_ was made, and that it contained test1.pbf and test1.lst. The .pbf file is the playbackfile and it will contain:

```
'' **CfMC**: /u/cfmctest/devel/playbackfiles/test1.p_/test1.pbf
'' **CfMC**: SURVENT 7.7 v.23aug05 (8:12Oct05)
'' **CfMC**: 13 OCT 2005 15:07 ldevs=1,8703,0, compile #772881
''>language character_set=Universal
#:time=13OCT05 15:07:30#
#:c->survmode=1#
#:(what to do=)#
'' interview with study test1, signature = 1cee6860,1 compiled at 13OCT05 15:06:58
#:(Q1:5.1)1#
#:nextid=2#
#:(what to do=)q#
'' **CfMC**: signature=2081700640
```

The first three lines log some information about the session. The commented out >language command records the character set you were using at the time of the interview. Time= shows the date and time; survmode= indicates the style of survent that was being run (e.g. screen, web/html, etc.), and "what to do=" is the "Press to interview" prompt. The "Q1" line contains the question label, the data location and width, and then the answer given. Nextid= is exactly that, and then, in this example we quit. This is the basic format of the contents of any pbf file.

Generally, the answers to questions in a pbf file are recorded just as they would be entered in screen mode. For example, a “^" will record a backup in screen mode or clicking the "previous" button in Web mode. One exception is that the end of text question input is a "*" by itself as an answer. For example:

```
#:(Q1:5.1)Here's the first line in the text box#
#:(Q1:5.1)and this is the second line, then we stop#
#:(Q1:5.1)*#  '' <-- This says we're all done.
```

The default file names for playback files use the study as their root. By default, a directory named study.p_ is created to contain the files.

stand-alone Survent	study.pbf
webSurvent	study_password.pbf
servered Survent	study_ldev.pbf
webCATI	study_interviewerid.pbf.

If you do not want a study.p_ subdirectory to be created, then add -STORE_FILES_IN_STUDY_DIRECTORY to your >playbackfiles command and the files will be created in the current directory.

If you would like the files to be created with a root name other than the study, then use BASE_FILENAME= on the >playbackfiles command.

The naming rules described above will be followed but using the value of base_filename instead of the study.

In addition to a pbf file, turning on >playbackfiles will cause a list file (lst) to be created. This will be a normal Survent list file for the session and can be very useful for debugging problems that occur behind the scenes while operating in web mode. The list file's name will follow the same naming pattern used for pbf files, making it very easy to tell which lst goes with which pbf.

If you would like to create a presentation file, include the MAKE_PRESENTATION_FILE option on the >playbackfiles command. The contents of the presentation file may be line mode, screen mode, or html. The type is controlled by the >echo_presentation command which takes the values HTML, SCREEN, or LINE_MODE.

HTML presentation files are a representation of all of the pages presented during the interview to the respondent, but strung together as if the interview were on a very long scroll. You should be able to look at one of these files with a browser and see a good representation of what the interviewer saw during the interview. You may need to give the file an html extension to get your browser to recognize the file as being html. If you need to do that, and you expect to be looking at these files frequently use ">cfmcextension prs=html" to make prs files have an html extension. HTML is the default for >echo_presentation.

SCREEN presentation causes a prs to be made with a listing of calls to write to the screen. These files are not very accessable, but do contain all of the text that was sent to the screen. This mode is most useful for testing that the screen presentation is remaining constant by comparing new screen presentation files with ones known to be good.

LINE_MODE gives a very simple text version of what was presented to the user. This can be a useful way of simplifying what went on in an interview when you are trying to sort out a complicated problem.

The important thing to realize is that with >echo_presentation set to HTML you can run stand-alone screen mode interviews and get prs files that will show you what the web user would see.

One thing that may be disconcerting at first when using prs files is that they contain what survent presented to the interviewer to be answered, and thus you will not see the answers to questions unless a backup occurred in the interview.

The ECHO_TELLS >playbackfiles option causes some additional parts of the interview dialog to be saved. The ADD_COMMENTS option causes a very large amount of information about what's going on during the interview to be recorded in the pbf file. In particular, c=#### is added to the lines indicating the "caller", which a programmer can use to help in debugging problems.

By default, the files created by >playbackfiles are appended to existing files of the same name. Thus, a webSurvent interview can suspend and, when resumed, it will continue using the original pbf file since the default name is study_password.pbf. In a shop where every interviewer uses their own ID this can also be useful also to collect all of any one interviewer's activity in a single file even if they've quit and restarted survent. (BUG: God knows what a mess you'll make if all the interviewers use the same ID and APPEND is in effect.) In some cases, however, appending to existing files may not be desirable, in which case the -APPEND option should be used on >playbackfiles.

A related option is SAME_LISTFILE. Turning this option on makes it so that if a list file is already open when survent starts a new study.lst will not be created but instead the run will continue with the existing list file. Usually this only makes a difference in test and debugging setups, especially those using "interview from_specs".

In order to replay a pbf file use a specfile like the following as input to Survent:

```
>playbackfiles basefilename=replay playbackmode=fullreplay
test1
dbug
y
&test1.pbf
```

Note that if you didn't use basefilename here this run would try to make a new test1.pbf, which is being ampersanded into the run, since that would be the default pbf name.

Usually you will want to set playbackmode to full replay because this will allow the replay to pay attention to such things as the random calls, and therefore be able to reproduce rotations etc.

The other mode is answers_only, which only uses explicit answers to questions and drives the interview forward somewhat blindly, much the same way a keyfile does.

When replaying an earlier interview if the question that the pbf attempts to answer is not the question that survent was expecting you will get an error in the form:

```
(ERROR #0) playback (ref=15387) expected match for
"#:(Q1:5.1)" but got input "#:(Q2:5.1)1#"
```

Depending on why you are doing a replay you may be happy or not about this. What this error means is that this trip through the interview isn't doing what it did when the pbf was made.

You can use the usual CfMC """ comments in pbf files and any of the meta commands that you would normally be allowed to use in Survent (you will have to be DBUG). In the case of making certain types of demos and tests, you can even use >repeat to do things, such as suspend and resume 100 times in a row.

4.7 SETTING THE CASE ID

The case ID identifies each completed record. You should not have duplicate case IDs because it is easy to get confused about what record belongs where. Sometimes case IDs are used to separate the data into groups; for instance each wave of a study might have case ids starting at 1001, 2001, etc.

CfMC uses a concept called “next case id” to control this. This value is stored in the data file and a copy of it is also stored in the quota file. It can be changed or set in three ways.

First, you may set it by creating or opening a file using the ~CLN FILE command or the “~INPUT <file> ALLOW_UPDATE” command and setting the NEXTCASEID=NNNN parameter; both of these commands can be used in the CLEANIT utility program or in MENTOR.

Secondly, you may set it using the “NEXT_ID” command in the questionnaire header. If the server finds that the "next" case ID in the data file or quota file is set to a value lower than the “NEXT_ID” in the QFF file, it sets them up to the higher value.

Thirdly, you may set it using the “MID” command in the supervisor. This sets the value in the data file and quota file. You can never set this lower than the prior value.

When the CfMC server runs, it will set the next case id to be the highest of these three settings.

This is to avoid creating duplicate case ids on respondent records. In standalone DOS Survent, the “next case id” value may also be set on the configuration screen or in the configuration file. To set the value in the file, edit SURVENT.CFG and place the case ID value on line 11 of the file. This allows users who send out diskettes to preset the group of case IDs that will be returned from that computer.

You *cannot* set the next case ID on the SURVENT.CFG menu until after the data file is created (the first time you run Survent). This is because the program looks to the QFF file and the QUO file to get the initial case ID the first time on the menu.

SURVENT UTILITY PROGRAMS

5

5.1 SURVENT UTILITIES

CfMC has an extensive library of system-wide functions and utilities. These are documented in the *Utilities* manual. The utilities documented in this chapter are specific to Survent users only. They are:

FIXRESUM

Updates suspended interviews so they can be resumed by a changed questionnaire, or tells you whether they may be resumed properly. If they cannot be resumed properly, it tells you where the changes that caused that are.

MAKECFG

Displays the list of active devices and/or studies under the CfMC server. You may clear devices if necessary.

MAKEPREP

Reads an ASCII questionnaire, typically from a word processor or editor, and converts it into a PREPARE specification file that can be used to create a questionnaire.

MAKEZONE

Lets you add area codes to an existing ZONETABL file on the fly (or you can pick up the most recent version from the CfMC distribution area).

RECODE

Lets you add, change, or recode data in an existing CfMC data (TR) file (single user only).

REFORMAT

Reads a questionnaire (QFF) file or variables (DB) file and a data (TR) file, and produces a spread (ASCII) data file, a map of the new locations and matching data definitions for tabulation packages.

QUOTAMOD

Reads a quota file (QUO), displays the quota values, and allows you to change the values. Can also add quota files together.

QUOTARPT

Reads a quota file and phone sample file and reports statistics on % of job done, time to completion and completion rates.

SUPRMENU/CFMCMENU/PROGMENU

Provides from simple to complete menus for the CfMC utility programs and some operating system functions.

SURVMON

Allows monitoring of interviews from outside SURVSUPR. Can control access to stations and rate the interviewer being monitored.

SURVVIEW

Allows you to VIEW and ALTER a completed interview; this has the same function as the "VIEW" command in Survent or the Supervisor, but has menus of studies and interviewers to choose from and has some other extra commands.

SUSPRES

Prints a list of the time and question suspended at, exports a separate CfMC (TR) data record for each suspended interview, and a second file with a list of all the question labels and their responses.

For complete documentation on all other Utility programs, see the *Utilities* manual. For your information, here is a short description of each of the utilities that are documented in the *Utilities* manual:

General Utilities

DBUTIL

Display information on variables DB files, fix them, or copy items from one DB file to another.

MAKEMSG

Makes the CfMC MSGFILE that contains all the program messages, allowing the user to change the wording or add languages to the messages.

Data Display Utilities

HOLE

Produces a summary report on the binary punches in the data, column by column, called a marginal or holecount. Allows bases, weights, and print controls.

LIST

Shows responses to open questions (VAR and TEX) or other questions across cases or within a case across questions.

SCAN

Produces cross-tabulations with counts and percentages across variables. Counts all values in string or numeric fields. Allows bases, weights, and a banner. Has print output, delimited output, or html output.

Data-Modifying Utilities

CLEANIT

Allows you direct access to the data in a CfMC data (TR) file to display and/or edit quickly and easily.

CODEEDIT

Used to code open-ends and edit verbatims in a systematic manner.

COPYFILE

Used to translate, sort, copy and otherwise manipulate data files.

MAKECASE

Converts ASCII or binary card image files into System (TR) files with valid case checking.

MERGE

Combines two files into one by matching on a key in each record of the files.

RAWCOPY

Recovers data (TR) files that have been structurally corrupted by reading cases one at a time out of the file and building a new one.

VERBEDIT

Used to edit verbatims in a systematic manner.

Note: Many data display and modifying utilities run under the Mentor program. Examplespecification files to run these utilities in batch mode are available in the \CFMC\SPX subdirectory or group SPX.CFMC.

There are also utilities specific to Survent's Telephone Number Management System. For information on these utilities, see *Chapter 6: TELEPHONE NUMBER MANAGEMENT SYSTEM.*

5.2 SUPRMENU/CFMCMENU/PROGMENU

These are utilities that provide a menu for the CfMC programs. They work on all supported platforms. You can modify the menus to meet your shop's needs.

There are three versions of the menu. SUPRMENU is the simplest version, the one that supervisors and the like will probably want to use. CFMCMENU has all the programs that a standard project manager might want to use, and PROGMENU has every relevant program.

CFMCMENU and PROGMENU have two options: one to use study files under the CfMC server, and one to use local files.

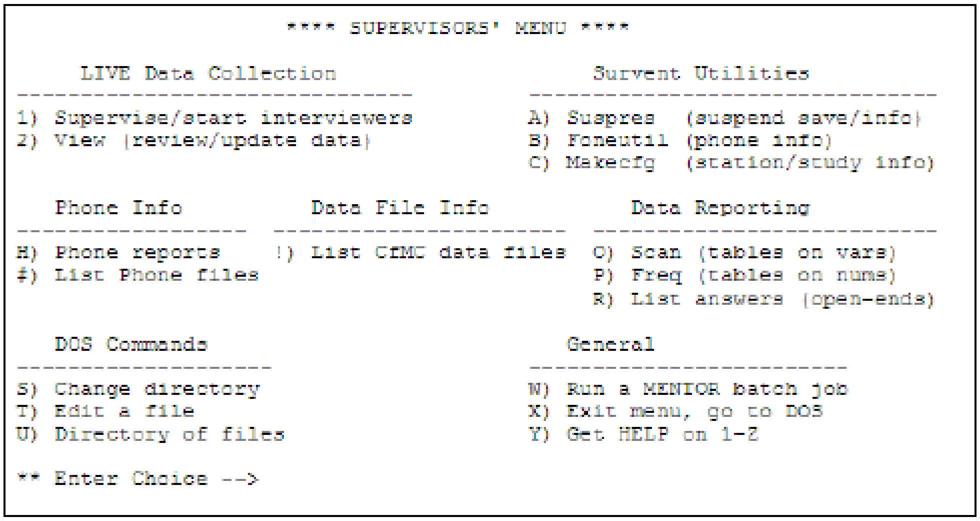
The benefits of using one of the menus include:

- Ease of getting new people using the software
- Not having to remember the program name
- Being able to run programs with help available

Here is the simplest version. It will only run on study files (studies that are running under the CfMC server). To run it, enter:

```
SUPRMENU
```

SUPRMENU



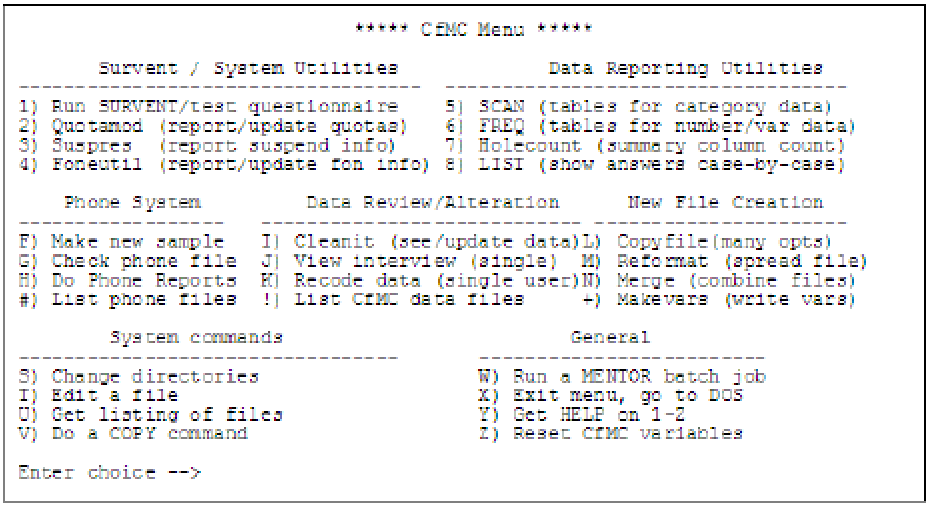
```
                        **** SUPERVISORS' MENU ****

     LIVE Data Collection                        Survent Utilities
 --------------------------------        ---------------------------------
 1) Supervise/start interviewers         A) Suspres  (suspend save/info)
 2) View (review/update data)            B) Foneutil (phone info)
                                         C) Makecfg  (station/study info)

   Phone Info           Data File Info             Data Reporting
 -------------------  -----------------------  ----------------------------
 H) Phone reports     !) List CfMC data files  O) Scan (tables on vars)
 #) List Phone files                           P) Freq (tables on nums)
                                               R) List answers (open-ends)

   DOS Commands                                 General
 ---------------------                   --------------------------
 S) Change directory                     W) Run a MENTOR batch job
 T) Edit a file                          X) Exit menu, go to DOS
 U) Directory of files                   Y) Get HELP on 1-Z

 ** Enter Choice -->
```

Option 1 runs the SURVSUPR program. Option 2 runs Survent in VIEW mode. Options 3-5are standard Survent utilities. Information on options H through R are listed in the *Utilities* manual. Look in this help screen (option Y) for more information on each item.

The standard project manager's menu is called "CFMCMENU". To run it, type:

```
CFMCMENU
```

CFMCMENU



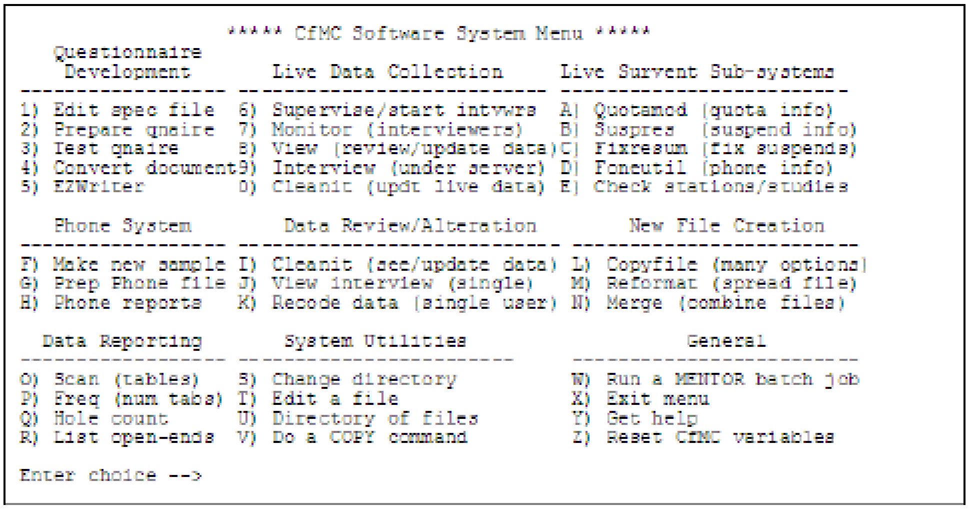
```
                         ***** CfMC Menu *****

      Survent / System Utilities            Data Reporting Utilities
-----------------------------------   -----------------------------------
1) Run SURVENT/test questionnaire     5| SCAN (tables for category data)
2) Quotamod (report/update quotas)    6| FREQ (tables for number/var data)
3) Suspres  (report suspend info)     7| Holecount (summary column count)
4) Foneutil (report/update fon info)  8| LIST (show answers case-by-case)

    Phone System         Data Review/Alteration      New File Creation
------------------   ---------------------------- ----------------------
F) Make new sample   I| Cleanit (see/update data)L) Copyfile(many opts)
G) Check phone file  J| View interview (single)  M) Reformat (spread file)
H) Do Phone Reports  K| Recode data (single user)N) Merge (combine files)
#) List phone files  !| List CfMC data files     +) Makevars (write vars)

        System commands                          General
---------------------------------         -------------------------
S) Change directories                     W) Run a MENTOR batch job
I) Edit a file                            X) Exit menu, go to DOS
U) Get listing of files                   Y) Get HELP on 1-Z
V) Do a COPY command                      Z) Reset CfMC variables

Enter choice -->
```

Look in this manual’s index or the help screen (item Y) for more information on each item. Items 5-8, I, L, N and + are described in the *Utilities* manual.

Here is the most complete version. This is called the rogrammers’ menu. From this menu, you can run almost any CfMC program. To run it, enter:

```
        PROGMENU
```

PROGMENU



```
                  ***** CfMC Software System Menu *****
   Questionnaire
    Development        Live Data Collection     Live Survent Sub-systems
------------------ --------------------------- -------------------------
1) Edit spec file  6) Supervise/start intvwrs  A] Quotamod (quota info)
2) Prepare qnaire  7) Monitor (interviewers)   B] Suspres  (suspend info)
3) Test qnaire     8) View (review/update data)C] Fixresum (fix suspends)
4) Convert document9) Interview (under server) D] Foneutil (phone info)
5) EZWriter        0) Cleanit (updt live data) E] Check stations/studies

   Phone System         Data Review/Alteration        New File Creation
------------------ ---------------------------- -------------------------
F) Make new sample I) Cleanit (see/update data) L) Copyfile (many options)
G) Prep Phone file J) View interview (single)   M) Reformat (spread file)
H) Phone reports   K) Recode data (single user) N) Merge (combine files)

  Data Reporting        System Utilities                  General
------------------ ------------------------     -------------------------
O) Scan (tables)   S) Change directory          W) Run a MENTOR batch job
P) Freq (num tabs) T) Edit a file               X) Exit menu
Q) Hole count      U) Directory of files        Y) Get help
R) List open-ends  V) Do a COPY command         Z) Reset CfMC variables

Enter choice -->
```

Look in this manual's index or the help screen (item Y) for more information on each item. Items O, J, L, and N-R and described in the *Utilities* manual.

5.3 MAKEPREP

MAKEPREP takes an ASCII file and makes a preliminary PREPARE spec file from it (makes CAT or VAR questions only, 132 characters maximum). This is basically a way to convert a word-processed version of the questionnaire into basic PREPARE specs. Make sure you keep the file as a flat ASCII file, not a word processor program file.

The only modification you'll have to make to the file is to insert an open curly brace ({) on a separate line before each question begins. A close curly brace (}) after the question is optional. After the last question, close it off by adding two lines: one with a close curly brace, and one more line with END on it to tell MAKEPREP you're done. Remove any extra blank lines in your file after the END line. If you want to mark response lists for CAT questions, precede the list with ~ or ~> (see below).

The syntax to run the program is:

```
    MAKEPREP infile listfile outspecfile
```

NOTE: You must specify all parameters from the operating system prompt. You will get an error if you specify file names at the Spec File/List File prompts, since this does not allow for the required outspecfile name. You may specify a dash before either or both of the listfile and outspecfile names to indicate purging of same-named files (instead of renaming).

Outspecfile name becomes the study name enclosed in brackets ([ ]) at the top of the file. Make this a 4- to 8-character name and do not include an extension or you will get an error when resuming the file in the Script Composer or compiling. No header options are put out, just the study name. Modify this yourself if necessary.

Rules:

- Question types are passed as VAR unless a recode table is indicated for CAT, or unless you specifically put in a !qtype line in which case it will be created as the question type specified. ***Note:*** Any line beginning with an exclamation point will be read as a question type line and will override the default type of VAR.
- Blank lines are passed to the outspecfile.
- A { in column 1 starts a new question and ends the previous question. As an option, you may indicate the end of a question with a closing brace (}).
- {- indicates a comment block and will be passed to the outspecfile as {!COMMENT. Nothing but {- may be on this line. {- like { ends the previous question, if any.
- After { in column 1, you may specify any of the parameters allowed on the question label line, such as label:, #, col.wid, HIDE, ALIAS=.

- A line with ~ in column 1 indicates that a recode table follows and the question will be passed as a CAT type to the outspecfile.

Options:

- ~> means that the response codes are to the right of the text and should be moved to the left in the outspecfile.
- ~>(special character) means look for this character in addition to a blank when looking for the start of the response code and do not pass any to the outspecfile.

Here is an example if your response list looks like this:

```
EX:
YES.......01
NO........02
```

you would put ~>. on the line above the response list in your file to start the recode table, move codes left, and indicate that the codes start after the character ".".

```
~>.
YES.......01
NO........02
```

and the outspecfile would look like this:

```
01 YES
02 NO
```

NOTE:

The length of the response code to move is determined by the first code found.

What you will get as an output from MAKEPREP is a PREPARE spec file that you can read into the Script Composer or open in an editor to have a specification-literate person modify to add logic conditions, change question types, etc. There will be a header statement at the front with the name of the outspecfile (so you probably either want to use a 4-8 character name for this or be prepared to modify the header). The END statement will be the last line.

You may want to delete this line or change it to ~END.

Any questions with large response lists should be marked as CAT questions before running MAKEPREP. If you don't, you will probably see errors about the question not fitting on the screen.

If you check for a good header and remove the END line, you will be able to make modifications in the Script Composer.

5.4 QUOTAMOD

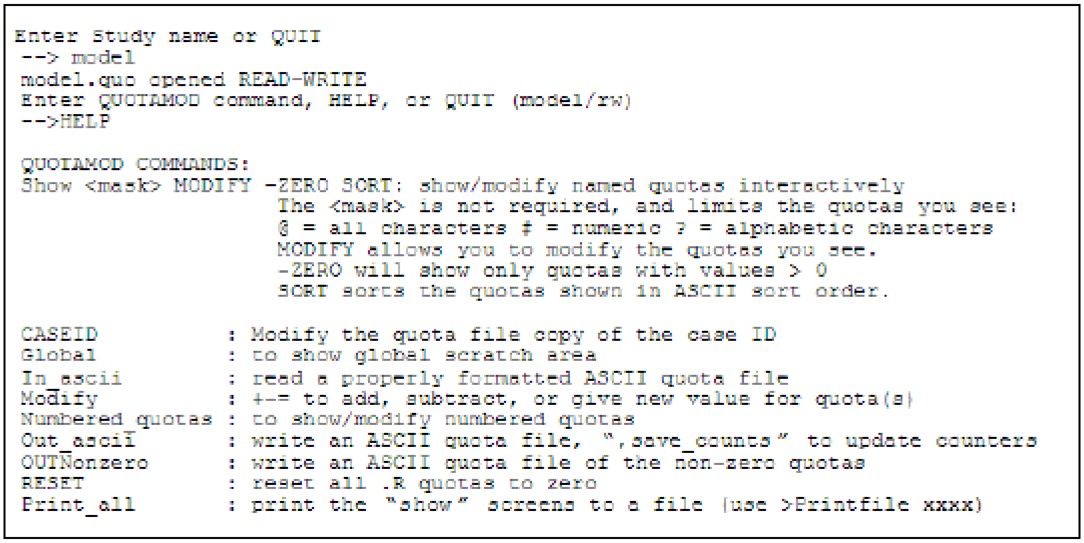
QUOTAMOD displays quotas or updates a quota file (QUO). It can also add separate quota files to a larger master quota file in the event that the network/host has crashed, or change the value of the master quota.

It looks at the quota file referenced by your SET CFMCQUOTA=name path (DOS, UNIX). Otherwise, it defaults to the local directory quota file. To use QUOTAMOD, at the operating system prompt, enter:

```
QUOTAMOD
```

You will get the standard Spec File and List File prompts. You can also specify CON CON or the specific file names on the program line. See *2.2 USING PREPARE* for more information on these prompts.

First, you are prompted for a study name. This is the name of the file holding the quotas. The file will have QUO as an extension, but you just enter the first part of the name. You can enter $filename if the file is elsewhere or use the CFMCQUOTA variable.



```
Enter Study name or QUIT
 --> model
 model.quo opened READ-WRITE
 Enter QUOTAMOD command, HELP, or QUIT (model/rw)
 -->HELP

 QUOTAMOD COMMANDS:
 Show <mask> MODIFY -ZERO SORT: show/modify named quotas interactively
                        The <mask> is not required, and limits the quotas you see:
                        @ = all characters # = numeric ? = alphabetic characters
                        MODIFY allows you to modify the quotas you see.
                        -ZERO will show only quotas with values > 0
                        SORT sorts the quotas shown in ASCII sort order.

 CASEID            : Modify the quota file copy of the case ID
 Global            : to show global scratch area
 In_ascii          : read a properly formatted ASCII quota file
 Modify            : +-= to add, subtract, or give new value for quota(s)
 Numbered_quotas   : to show/modify numbered quotas
 Out_ascii         : write an ASCII quota file, ",save_counts" to update counters
 OUTNonzero        : write an ASCII quota file of the non-zero quotas
 RESET             : reset all .R quotas to zero
 Print_all         : print the "show" screens to a file (use >Printfile xxxx)
```

Choose an option by entering the upper-case letter(s).

CASEID Changes the ID in the quota file. This is the ID that would be used if the data file case ID is not already set higher.

G Global scratch area is displayed. SPC,M and N statements put things into and take things out of the global scratch area.

```
EX:
     0         1         2         3         4
     1234567890123456789012345678901234567890
     Info from the global scratch area
```

I In_ASCII lets you read in an ASCII file of quota statements with changes to the quota file.

The changes should be in the format of:

```
<quota name> [+-=] <number> (named quota)
EX:
Males +5
```

```
Females=50
#<quota name> [+-=] <number> (numbered quota)

EX:
#152 -5
#232=275
```

+<number> adds to the current quota value

-<number> subtracts from the current quota value

=<number> changes the current quota value to the number specified here.

= is the default and will be used if no +/-/= is specified.

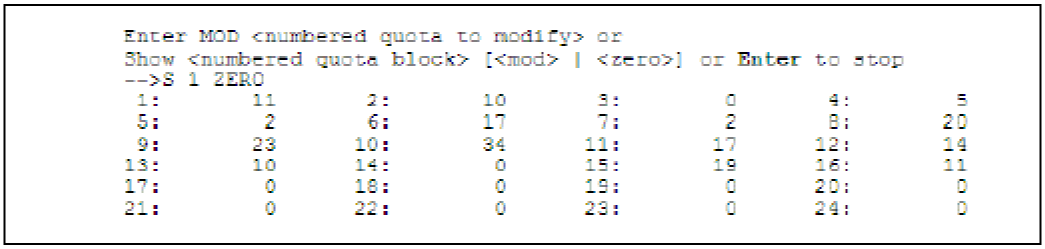
The file should have one change per line and should end with a line with END on it.

```
EX:
-->I
Filename from which to import quotas or Quit
-->QCHANG
Bank card +2
Genpurp -5
Retail=175
survent_totaltime=15:43:20
END
```

M Modify lets you change the current quota values on quotas. See the S option or N option to modify from a "menu". Use the same syntax as In_ASCII option, or use a quotaname pattern to modify many quotas. Name patterns may use * (all), ? (letters or numbers), or # (numbers).

```
<quota name> [+-=] <number> or
#<numbered quota> [+-=]<number> or
END
-->Bluecoll +10
-->Group*.T=175
-->#257 -2
-->END
```

N Numbered quotas displays the current quota values for numbered quotas. You will be prompted to Show or **Enter**. If you choose Show, you must specify a block of numbered quotas to show, or ZERO to show filled and zero quotas also.



```
Enter MOD <numbered quota to modify> or
Show <numbered quota block> [<mod> | <zero>] or Enter to stop
-->S 1 ZERO
 1:        11       2:        10       3:         0       4:         5
 5:         2       6:        17       7:         2       8:        20
 9:        23      10:        34      11:        17      12:        14
13:        10      14:         0      15:        19      16:        11
17:         0      18:         0      19:         0      20:         0
21:         0      22:         0      23:         0      24:         0
```

When you enter “S 1 ZERO” you will see a display beginning with the first numbered quota. You could enter “S 15 ZERO” which displays quotas beginning with quota number 15.

The number used for your MAXIMUM_QUOTA_NUMBER value in your header statement will limit what you can see here.

MOD allows you to modify the quotas on the screen. Just move to the appropriate quota field (by using the arrow keys on monitors, and **Ctrl-U**, **D**, **L**, or **R** on terminals) and enter the replacement value.

ZERO will display quotas with no current value. The default is that they are not shown.

O Out_ASCII outputs an ASCII file with the current quota names and values. You are prompted for the filename; the default name is studyAQU. If you wanted to make many changes to your quota values, use the OUT_ASCII option, modify the AQU file in the editor, then use the IN_ASCII option to read those changes back in. If you use the option OUT_ASCII,SAVE_COUNTS, then the QUOTAMOD variables will be saved such that when you import to a new file it will update their values. See *Using QUOTAMOD Variables* below for a discussion on using the OUT_ASCII option to set system variables. ***NOTE:*** You could also use the SHOW/MODIFY option to make quota changes.

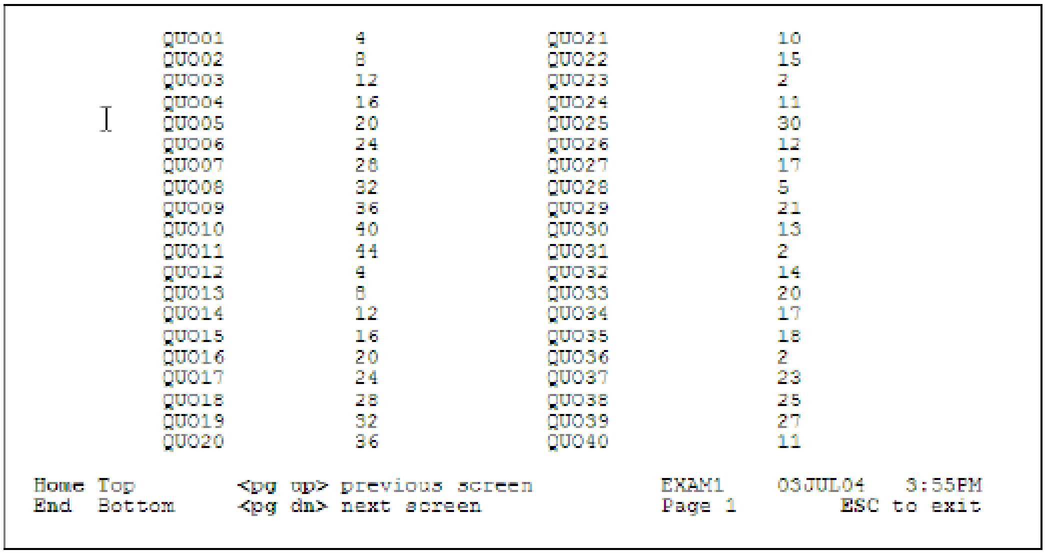
OUTN OUTNonzero is the same as OutASCII except it only outputs the nonzero quotas.

P Print_all prints the show quota screen(s) to the currently opened >PRINT_FILE or LIST FILE. If you say P <filename> (i.e., “P QLIST”), it prints to the specified file.

Q Quit this study. The beginning prompt “Enter Study name or QUIT” will again appear, giving you a chance to work on another study.

RESET Reset all .R quotas to zero. These quotas are created when in Triplequota mode. These can be used to show the filling of quotas on some regular basis, then reset at the start of the next period (day, week, etc.).

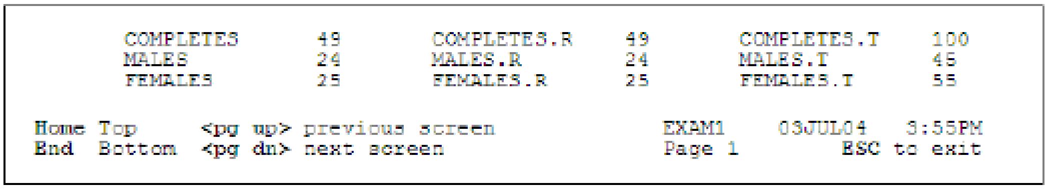
S Show quotas will display current quota values for named quotas and allows you to modify quotas.



```
      QUO01          4            QUO21          10
      QUO02          8            QUO22          15
      QUO03          12           QUO23          2
      QUO04          16           QUO24          11
      QUO05          20           QUO25          30
      QUO06          24           QUO26          12
      QUO07          28           QUO27          17
      QUO08          32           QUO28          5
      QUO09          36           QUO29          21
      QUO10          40           QUO30          13
      QUO11          44           QUO31          2
      QUO12          4            QUO32          14
      QUO13          8            QUO33          20
      QUO14          12           QUO34          17
      QUO15          16           QUO35          18
      QUO16          20           QUO36          2
      QUO17          24           QUO37          23
      QUO18          28           QUO38          25
      QUO19          32           QUO39          27
      QUO20          36           QUO40          11

Home Top       <pg up> previous screen        EXAM1     03JUL04    3:55PM
End  Bottom    <pg dn> next screen            Page 1         ESC to exit
```

NOTE: UNIX users will be prompted with **Ctrl-G** (top), **Ctrl-E** (End), **Ctrl-T** (next), and **Ctrl-V** (previous).

If you are using Triplequotas mode, the display will look like this:



```
     COMPLETES       49        COMPLETES.R    49        COMPLETES.T    100
     MALES           24        MALES.R        24        MALES.T        45
     FEMALES         25        FEMALES.R      25        FEMALES.T      55

Home Top      <pg up> previous screen              EXAM1    03JUL04    3:55PM
End  Bottom   <pg dn> next screen                  Page 1         ESC to exit
```

This allows you to change your target quota, without resetting the actual quota value, if you decide to have more or less of a particular group.

<mask> optional; if used, will limit the quotas you see. The mask can include the following characters:

* or @ all characters, however many

# numeric characters; one per #

? alphabetic characters; one per ?

```
EX:
SHOW TAR*
```

Displays all quotas beginning with TAR and ending with anything else (or nothing else). This will match: TAR, TARGET, TAR_G1, etc. SHOW MALE## Displays all quotas beginning with MALE and ending with exactly two numbers. This will match: MALE01, MALE99, etc.

```
SHOW DOG?
```

Displays all quotas beginning with DOG and ending with exactly one letter. This will match: DOGS, DOGY, etc.

MODIFY Allows you to modify the quotas displayed, by highlighting the quota wanted and typing in a new value. The highlighting is done by moving on the screen using the arrow keys (monitors) or **Ctrl-U**,**D**,**R**,**L** (terminals) to move up, down, right, and left.

-**ZERO** Display only nonzero quotas.

SORT Display the quotas in sorted ASCII order

Using QUOTAMOD variables

The QUOTAMOD OUT option writes 5 lines at the bottom of the OUT file in addition to the quota values. Here are the lines written:

```
EX:
```

```
*survent_totaltime = 002:55:26        Total time logged on study
*survent_currentime= 000:23:15        Time logged on in current session
*interviewing_totaltime  - 002:01:10  Total time in interviews
*interviewing_currentime - 000:18:54  Current time in interviews
*next_caseid = 0179                   Next case id to be assigned
*unique_number = 123                  Current counter value
```

The four times are totals across all interviewers; for example, if 4 interviewers each were logged onto the study for 2 hours, the total time would be 8 hours even if they were all logged on at the same time. These values can also be seen by using the"STUDY <studyname>" command in SURVSUPR. The QUOTARPT utility uses these values to calculate the time on the study and time left to finish.

The “next_caseid” variable is the value to be assigned to the next data record collected by Survent. This also can be modofied from the Supervisor uring the "modid" command. The “unique_number” variable has the value of the unique number assigned by the “!SPC,A,N” statement. It is used to guarantee that each case gets a unique number independent of the case id.

If you use the Quotamod OUT,SAVE_COUNTS option when exporting the .AQU file, the variables are written without the “*” character in front of their name. If you then import this file with the IN option, the values are reset to whatever you specify. You can also just edit the file and remove the “*” of items you want to be updated. For example, saying "Survent_totaltime = 010:00:00" will set the value of survent_totaltime to 10 hours. You might need to do this if you were re-compiling and mistakenly deleted the QUOTA file.

5.5 QUOTARPT

QUOTARPT is a report that combines information from the quota file (QUO) and phone sample file (FON) to produce an overview of how far along in a project you have gotten. It is most useful for studies using TARGET or DOUBLE quotas: it prints the current value and target value, the % complete, the number of interviews needed to complete the quota, the number of completes in the group per hour and the number of interviewer hours needed to finish the quota up. This should allow you to schedule the correct number of interviewers on the study to complete it according to schedule.

QUOTARPT is very fast because it operates on summary numbers in the QUOTA and PHONE files instead of reading through the whole file. Reports are generated in less than 30 seconds on most systems.

To run QUOTARPT, you may specify any of the following:

QUOTARPT <study> Runs a report on the study in question

QUOTARPT <study> LOCAL Runs a report using local files instead of files in the CfMC system area

QUOTARPT ALL Runs a report on all studies in the study area

QUOTARPT USELIST Runs a report on a list of studies you maintain If you say “QUOTARPT <study>” the program runs a report on the study specified. The report is displayed on the screen and stored in the file <study>^QRT. “QUOTARPT <study> LOCAL” will do the same operation on files in the local directory instead of in the CfMC live study area.

“QUOTARPT ALL” runs reports on all files found in the CFMC quota directory. This will also write a <study>^QRT file for each study that it runs on, but will not put the reports on the screen. Be sure to delete old studies if you don’t want them to continue to be reported on.

“QUOTARPT LIVE” runs on studies currently being accessed by the CfMC SERVER. This is useful because you can set up a job to run every 10 minutes or every hour to provide updated values throughout the day without having to re-run the QUOTARPT program.

“QUOTARPT USELIST" says to use the file called USELIST in the CFMCFONE directory. You would have the studies you want reported on listed, 1 per line. This is the same file used by the FONERUN program to keep phone files updated daily.

Here is a sample of the report format using a small phone file and a quota file with target quotas:

```
QUOTA/PHONE REPORT FOR STUDY:  ***  PHONE  ***  15 SEP 2000 15:49      PAGE 1

                               COMP- PRCNT  RESET BALNCE CMPLETE RESLVD  #S TO HRS TO
QUOTA NAME        TARGET  LETES  COMP  -ABLE NEEDED  /HOUR   /COMP FINISH FINISH
---------------- ------ ------ ----- ------ ------ ------- ------ ------ ------
COMPLETE        :     20      2   10%      2     18    66.7    1.0     18    0.3
FEMALE          :     10      2   20%      2      8    66.7    1.0      8    0.1
MALE            :     10      1   10%      1      9    33.3    2.0     18    0.3
---------------  ------ ------ ----- ------ ------ ------- ------ ------ ------
OVERALL(Trg>0):      40      5   13%      5     35   166.7    0.4     14    0.2
HIGHEST VALUE :      20      2   20%      2     18    66.7    2.0     18    0.3
LOWEST VALUE  :      10      1   10%      1      8    33.3    1.0      8    0.1
---------------  ------ ------ ----- ------ ------ ------- ------ ------ ------
ANSWMACHS       :      0     12              0
BUSYS           :      0      3              0
CALLBACKS       :      0      4              0
TERM_IN_PROG    :      0      4              0

TOTAL    <-AVAILABLE-> RES-          ALL TRG| HOURS ON   HOURS IN   TOTAL  DIALS
SAMPLE    FRESH  CALLED OLVED HIDDEN ATTMPT|   STUDY    INTRVIEWS  DIALS  /HOUR
------- ------- ------ ------ ------ ------ ---------- --------- ------ -----
     26      23      1      2      0      0       0.03      0.03      3 100.0
```

The quotas with targets are listed first; they are sorted in “TARGET SIZE” order, so that the largest quotas are listed first. After the quotas with targets are listed, the program reports the sum of the quotas, the highest value seen in each column, and the lowest value seen.

Following that is a list of the quotas that had no target or whose targets are set to “0”; these will not have statistics listed, just counts.

After the quota information, information from the phone file is listed. This includes the total sample in the file, how many have been called, resolved, or hidden. Also listed are the hours on the study and hours in the interview from the quota file variables, then the total number of dials on the study and number of dials per hour.

This is the optimal report run on a study with a phone file and target quotas. You can also run on studies using “Double” or ordered “Numbered” quotas, but this requires modification of DEFINE statements at the top of the file QRPT^SPX in the CFMC SUPPORT directory. Youcan modify this file and save it locally, and QUOTARPT will use the modified file if you run reports there.

You can also run QUOTARPT on studies with no phone file or with single quotas without any spec file modifications, but you will only get the information the program can ascertain without target quotas or phone sample file information.

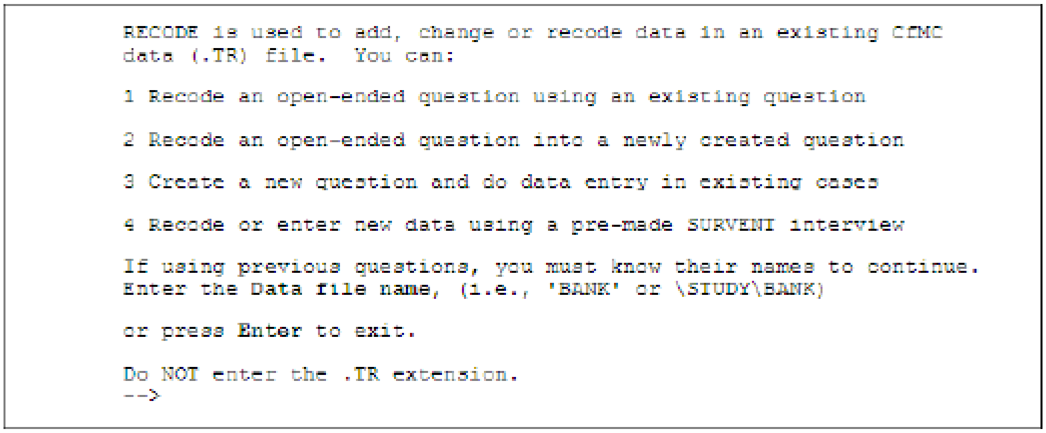
5.6 RECODE

The RECODE utility lets you modify or add data to a CfMC data (TR) file. It can also be used to code open-ends from Survent interviewing (see SPC,P or View mode for other ways to do this). Only one person can modify at a time. Use Coding mode or View mode for more than one person at a time.

To run RECODE, enter:

```
RECODE
```

and you will see the starting screen:



```
RECODE is used to add, change or recode data in an existing CfMC
data (.TR) file.  You can:

1 Recode an open-ended question using an existing question

2 Recode an open-ended question into a newly created question

3 Create a new question and do data entry in existing cases

4 Recode or enter new data using a pre-made SURVENT interview

If using previous questions, you must know their names to continue.
Enter the Data file name, (i.e., 'BANK' or \STUDY\BANK)

or press Enter to exit.

Do NOT enter the .TR extension.
-->
```

After entering the name of your data file, you will be asked to specify which of the four operations you want to do.

1 **Recode** open-end using an *existing/modified* code list.

If you choose 1), you will be prompted for a base definition. The base will determine who gets data entered for them in this run. Enter the base definition in the box shown on the screen.

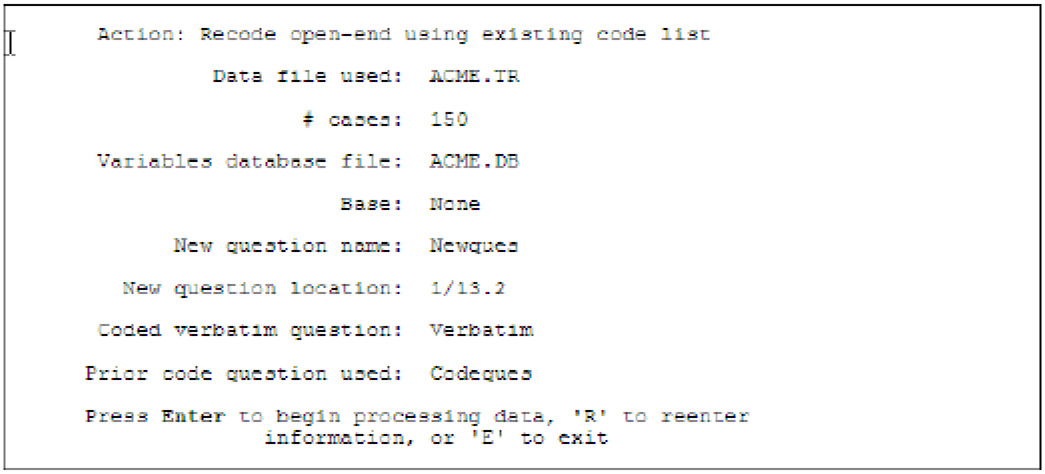
If you enter enough characters, you will be prompted with a second box to enter the rest (if any) of the base definition.

RECODE then asks if you have some pre-defined variables to use. If yes, you must enter thename of the DB file to get variables out of.

Now you're asked to supply a name for the new question you are creating, and then a data location and length for that question.

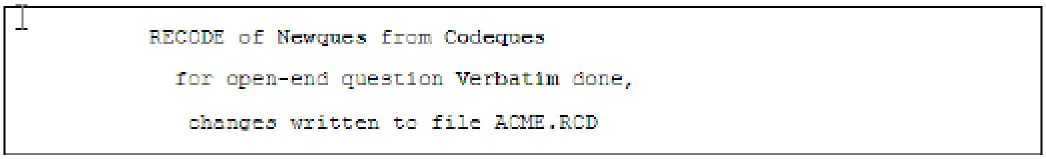
You're then asked for the label or question number of the original CAT or FLD question that has the categories to recode, and a description of the open-end response to be coded. This description can consist of a column location and length, a label or a question number.

You can then review the choices you've made.



```
      Action: Recode open-end using existing code list

               Data file used:  ACME.TR

                      # cases:  150

      Variables database file:  ACME.DB

                         Base:  None

            New question name:  Newques

        New question location:  1/13.2

      Coded verbatim question:  Verbatim

     Prior code question used:  Codeques

     Press Enter to begin processing data, 'R' to reenter
                   information, or 'E' to exit
```

RECODE will print some messages as it does its work, ending with this message before the program ends:



```
RECODE of Newques from Codeques
  for open-end question Verbatim done,
   changes written to file ACME.RCD
```

2 **Recode** open-end into a *new* code list.

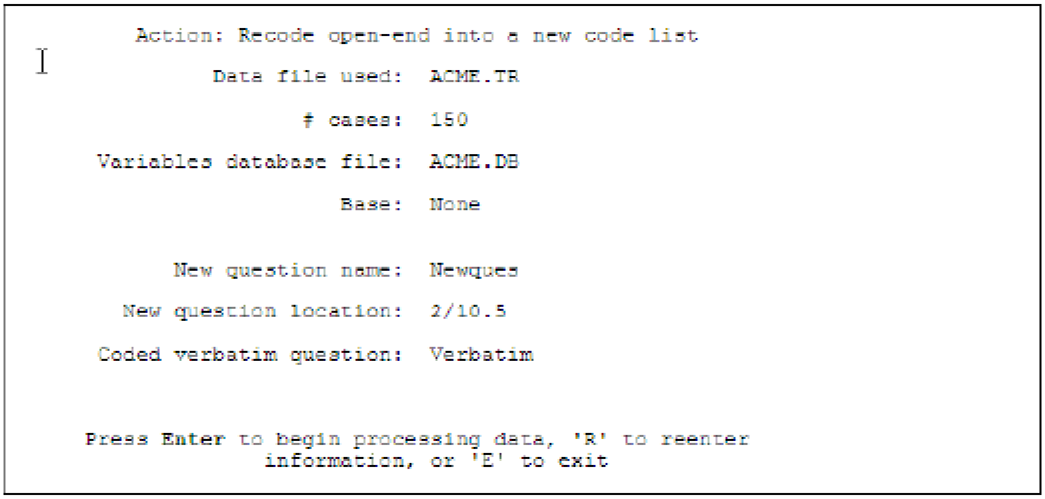
If you choose 2), you are prompted for a base definition. The base will determine who is recoded in this run. Enter the base definition in the box shown on the screen. If you enter enough characters, you will be prompted with a second box to enter the rest (if any) of the base definition.

RECODE then asks if you have some pre-defined variables to use. If yes, you must enter the name of the DB file from which to get the variables.

Now you're asked to supply a name for the new question you are creating, and then a data location and length for that question.

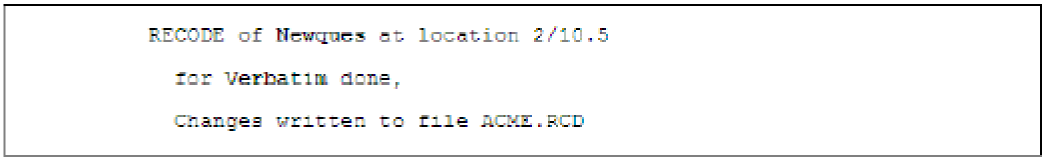
You are then prompted for a description of the open-end response to be coded. This description can consist of a column location and length, a label or a question number.

You can then review the choices you've made:



```
        Action: Recode open-end into a new code list

                 Data file used:  ACME.TR

                        # cases:  150

       Variables database file:  ACME.DB

                          Base:  None

            New question name:  Newques

        New question location:  2/10.5

       Coded verbatim question:  Verbatim


   Press Enter to begin processing data, 'R' to reenter
                 information, or 'E' to exit
```

RECODE will print some messages as it does its work. When it is done, it will display the following message and then end the program.



```
        RECODE of Newques at location 2/10.5

          for Verbatim done,

          Changes written to file ACME.RCD
```

3 **Create** a new question and do data entry.

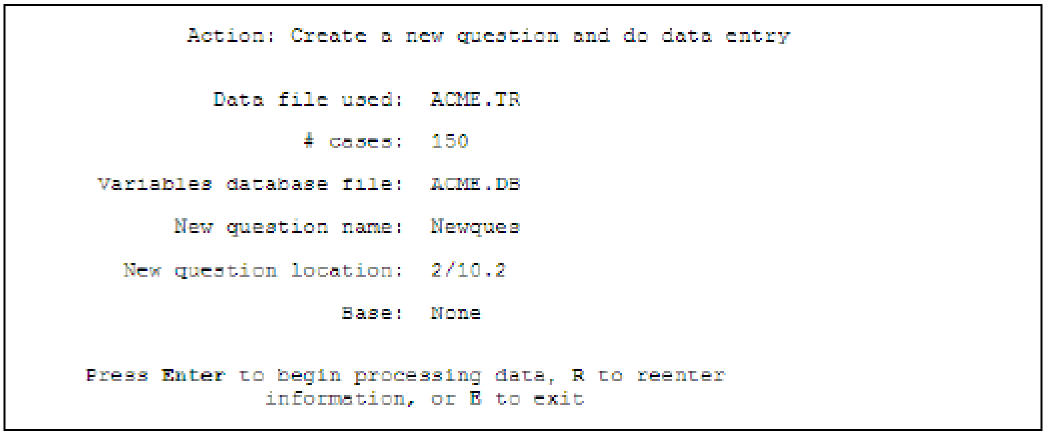
If you choose 3), you will be prompted for a base definition. The base will determine who gets data entered for them in this run. Enter the base definition in the box shown on the screen.

If you enter enough characters, you will be prompted with a second box to enter the rest (if any) of the base definition.

RECODE then asks if you have some pre-defined variables to use. If yes, you must enter the name of the DB file from which to get the variables.

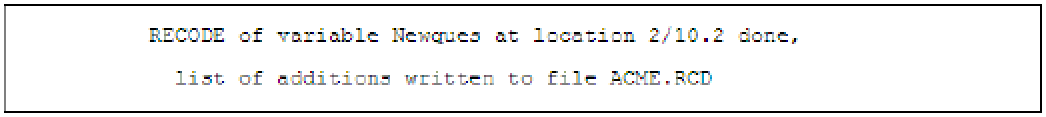
Now you're asked to supply a name for the new question you are creating and then a data location and length for that question.

You can then review the choices you've made:



```
              Action: Create a new question and do data entry

                   Data file used:  ACME.TR

                          # cases:  150

          Variables database file:  ACME.DB

                New question name:  Newques

            New question location:  2/10.2

                             Base:  None

     Press Enter to begin processing data, R to reenter
                   information, or E to exit
```

RECODE will display some messages indicating the work it is doing for the new question, followed by a final recap of what was done before the program ends.



```
          RECODE of variable Newques at location 2/10.2 done,

            list of additions written to file ACME.RCD
```

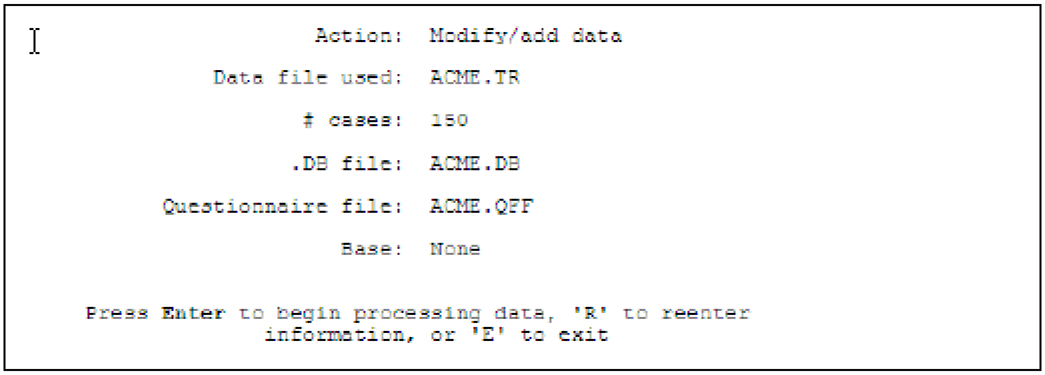
Recode or enter new data using an existing questionnaire If you choose 4), you will be prompted for the name of the questionnaire file to use, and if you want to use existing cases or add new cases to the data file.

If you choose to modify existing cases, you will be asked if you wish to pause between interviews and for the DB file from which to get the variables.

Then you are prompted for a base definition. The base will determine who is recoded in this run. Enter the base definition in the box shown on the screen. If you enter enough characters, you will be prompted with a second box to enter the rest (if any) of the base definition.

RECODE then asks if you have some pre-defined variables to use. If yes, you must enter the name of the DB file from which to get variables.

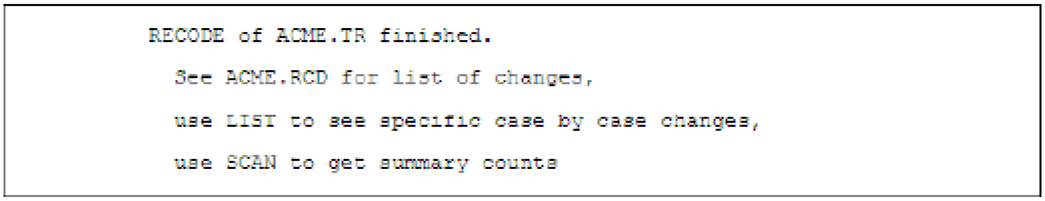
Now you will be given a chance to review the choices you've made so far.



```
                        Action:  Modify/add data

                Data file used:  ACME.TR

                       # cases:  150

                     .DB file:   ACME.DB

           Questionnaire file:   ACME.QFF

                         Base:   None

    Press Enter to begin processing data, 'R' to reenter
                 information, or 'E' to exit
```

Upon pressing **Enter**, you will see the message "... Please wait, building 'recode interviews' procedure" if you chose to modify/add data to existing cases or " ... Please wait, building 'add interviews' procedure" if you chose to add new cases to the file.

When the recoding is done, you see this screen:



```
RECODE of ACME.TR finished.

  See ACME.RCD for list of changes,

  use LIST to see specific case by case changes,

  use SCAN to get summary counts
```

5.7 REFORMAT

REFORMAT is a utility program that has the following uses:

- Reformatting data files from CfMC compressed column binary to ASCII format.
- Expanding CAT questions (spreading multi-punches).
- Generating other program language specification files to match the reformatted data.
- Converting TEX questions into ASCII data in conjunction with your other data types.
- Producing a delimited ASCII data file for spreadsheets.

REFORMAT produces two or more files:

- A spread ASCII data file (with an extension of RFT), which may be used as input to programs that use ASCII format. This may be a delimited file, fixed format or cardimage format.
- A map file (RFL), which shows the variable text and codes, where the data was moved to and, optionally, where it was moved from. This fixed format map can also be used to programmatically generate code to import the data to other packages.
- A definitions file (DEF) for data definitions to export to other data processing packages. For instance, using the keyword SSS_XML, REFORMAT will export data definitions in "Triple S" XML format, a popular data format for market research.

REFORMAT needs a data file (usually a CfMC TR file) and a questionnaire file (QFF) or variables file (DB) as input. The data file is usually the TR file created while interviewing with Survent. The questionnaire file is usually the QFF used to create the TR file, and the variables DB file would be the one created at the time the questionnaire was compiled. The questionnaire specification file may contain compiler commands to control the creation of the spread data file (see *3.2.4 DATA CONTROL COMMANDS, REFORMAT Data Controls and Use*). In addition, the REFORMAT utility allows you to choose from several options that will affect the format of both the RFL and RFT files. Some of these options override compiler commands in the questionnaire specification file.

Additionally, you may create a QFF or DB file simply as a data definition vehicle and spread any type of file (ASCII, binary, or CfMC FON file).

NOTE: You can produce a map of the data with only the QFF file.

In addition to the RFL and RFT files, you can generate any other CfMC file to match the newly reformatted data (e.g., DEF). You can generate Survent or Mentor specification files. Also, you can generate files for other data processing systems such as COSI, SAS or SPSS.

REFORMAT will process any questions that put something in the data. Here is a list of Survent question types that will be reformatted:

CATegory

EXPression

FieLD

LOOP

NUMeric

PHONE subtypes: G and T.

SPC subtypes: 3,4,6,7,9,F,L,N, and V

TEXt

VARiable

REFORMAT is a menu-driven program. Self-documenting screens guide you to produce both a map and a spread data file, or only the spread data file or map file. You are also offered several options (described below) that will affect the data and map file formats. A final review or recap screen allows you to change previously selected options before the map and spread files are made.

To use the program, at the operating system prompt, enter:

```
REFORMAT
```

Or enter the proper code from one of the CfMC Menu programs. First, you are asked the name of the data file to use. Most likely your file will be a standard CfMC TR data file, since if you have a questionnaire you have usually collected the data in a TR file using Survent.

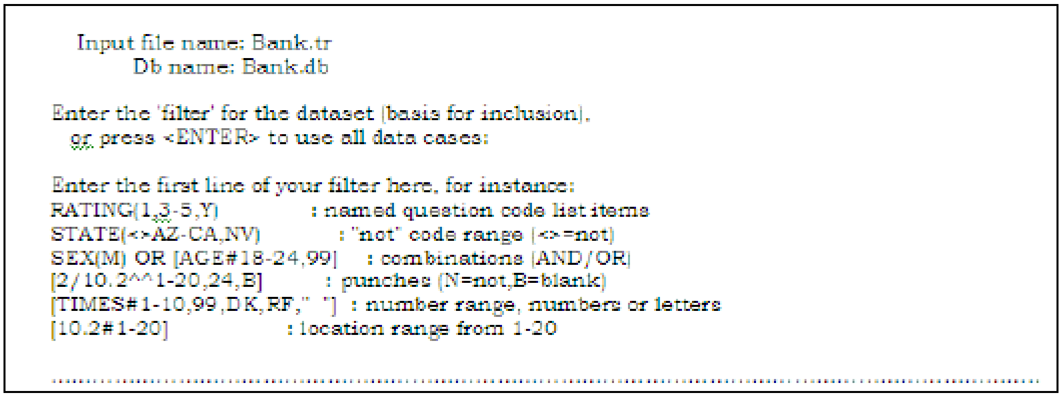
Then you are asked whether you are using the questionnaire file or a variables file for the reformat. Use the Questionnaire file if you are exporting all the variables, or if you have specifically set it up with the special compiler commands to control the reformat. Use the Variables DB file if you will just be doing a few variables that you want to list or give a subset of the variables to do. You can also use the DB file to get variables to use as a filter so that when you reformat the data you only use certain cases.

If you choose to use a questionnaire file, you will be prompted for a questionnaire file name.

The data is spread in the order that variables appear in the questionnaire. If you wish to change the order of the data spread, you must change the order of the questions in the questionnaire and recompile. Also, compiler commands can be placed in the questionnaire and after recompilation can further control the type of data spread (see below). Skip logic and conditional statements are ignored when REFORMAT spreads the data set; only the basic variable is used.

If you choose to use a variables DB file, you will be prompted for its name.

The program will then prompt for a FILTER to subset the records to be exported. If you have provided a DB file name, you can use the variables in the DB file to describe the filter, otherwise you will need to use data location references to describe the filter. This is the same screen seen in other CfMC utilities like SCAN and LIST that ask for a filter:



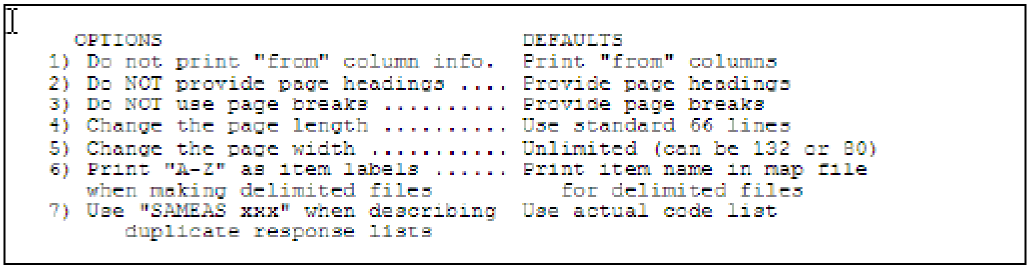
```
   Input file name: Bank.tr
         Db name: Bank.db

Enter the 'filter' for the dataset (basis for inclusion),
  or press <ENTER> to use all data cases:

Enter the first line of your filter here, for instance:
RATING(1,3-5,Y)              : named question code list items
STATE(<>AZ-CA,NV)            : "not" code range (<>=not)
SEX(M) OR [AGE#18-24,99]     : combinations (AND/OR)
[2/10.2^^1-20,24,B]          : punches (N=not,B=blank)
[TIMES#1-10,99,DK,RF," "]  : number range, numbers or letters
[10.2#1-20]                  : location range from 1-20

.................................................................................................
```

The next screen asks for the name to use for the output files. This can be from 1-8 characters in DOS or 1-14 characters in UNIX. This is the name the .rft (data), .rfl (map), and .def (data definition) files will be created with.

If you are using a variables DB file for the reformat, you will now be prompted for which variables you want to use. You can list specific variables in the order you want them, or you can ask for a subset of the variables based on variable type, variable name, or variables in a particular section of the questionnaire, eg. from question 23 to question 35. If you use specific variables, you can enter up to 20 variables in whatever order you wish them to be exported. NOTE: These are the same screens seen when requesting variables in the SCAN or LIST utilities.

The next screen will be the Map file options screen. It displays information about the options as follows:

The REFORMAT map (RFL) file prints all the information about the variables being reformatted. The map file is written in a fixed format to allow you to write programs that read it and create programs or documents using fixed position references. Here are the map file defaults and options:



```
  OPTIONS                             DEFAULTS
1) Do not print "from" column info.  Print "from" columns
2) Do NOT provide page headings .... Provide page headings
3) Do NOT use page breaks .......... Provide page breaks
4) Change the page length .......... Use standard 66 lines
5) Change the page width ........... Unlimited (can be 132 or 80)
6) Print "A-Z" as item labels ...... Print item name in map file
   when making delimited files          for delimited files
7) Use "SAMEAS xxx" when describing  Use actual code list
      duplicate response lists
```

The column on the left displays the options you can choose from, and the defaults are on the right.

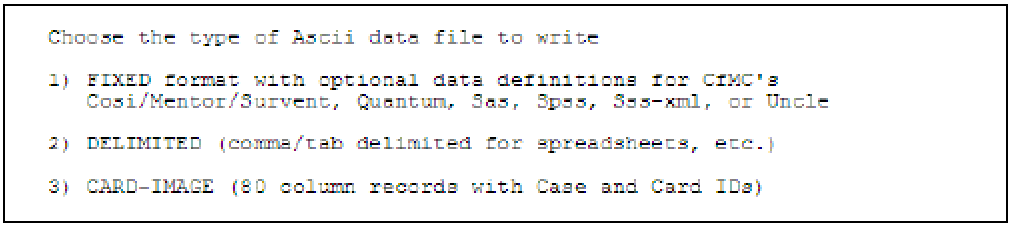
Choose all the items you want to change. Here is a brief description of the options:

- By default, the map file includes information about where the data came from and where it was spread to. If you wish to create a listing for someone that only includes where the data was spread TO, choose this option.
- By default, the program provides page headings with the name of the study, the page number, and column headings for the information listed for each question. If you want to exclude this information, choose this option. This may be useful to do, for instance, if you had a program to read the map file output and create variables for some program whose data definitions are not supported by CfMC.
- If you do not want to have page breaks in the map file, choose this option. This would be useful if you had a printer that could not properly print the pages with page breaks.
- Use this option to change the page length from the standard 66 lines per page.
- Changes the page width, you may want to wrap the map at 80 or 132 for printing portrait or landscape mode.
- By default, the program prints the variable names at the top when writing delimited files. Choose this option to use "A"-"Z" type names like those used by some spreadsheets by default.

- The map file includes a complete code list for every code list question by default. If you choose this option, the program will write one line with the notation "This uses the same code list as question XXX" for any question that used the SAMEAS feature in the questionnaire to use the same code list as a prior question.

Specifying the type of ASCII Data File to Create

Now we tell the program which type of file to write. This is dependent on the client or application you will be using the file for. Here is what the screen looks like:



```
Choose the type of Ascii data file to write

1) FIXED format with optional data definitions for CfMC's
   Cosi/Mentor/Survent, Quantum, Sas, Spss, Sss-xml, or Uncle

2) DELIMITED (comma/tab delimited for spreadsheets, etc.)

3) CARD-IMAGE (80 column records with Case and Card IDs)
```

4 Fixed format ASCII data spreads the data in the same relative position for every question across respondents. Records will be as long as necessary for the data spread. When CfMC writes data definitions for other packages, it must have the data in this format to match.

5 Delimited data is spread with a varying width (blanks stripped) for each question separated by the delimiter of your choice between variables. The variable names are recorded in the first record. You can choose the delimiter, common delimiters are TAB, comma, and blank. Quotes are placed around variables with string data, thus data from SPC subtypes 3, 4, 7, 9, and V; PHONE subtype G; and VAR is quoted. Numeric and code data is recorded without quotes around it.

6 Card-image data generates a case ID and record ID for each 80-column record generated, and then the data that fits on that record. By default, the record IDs will be numbered sequentially

from 01 to nn for the number of records created for each respondent; you may override this to allow more records by saying “CARD_IMAGE:3” which forces record IDs to start at 001 and go to 999. The same number of records are written for each respondent regardless of the amount of data in their particular data record. Record descriptions in the MAP file are written in record number/column format (e.g., 2/23 means record number 2, column 23, or absolute position 103). You can specify the width of a record ID, naming a value from 1-9. If the width you specify is too short, REFORMAT will put asterisks (*) into the 10 field instead of a value for those records requiring the larger space.

Creating a Data Definition File to Match the Reformatted Data

A popular option is to spread the data in fixed format and produce a matching set of data definitions in some other command language for data processing. If you have CfMC's Mentor program, you can use that to process the data using the standard CfMC TR file, but if you do not, or you are sending the data to someone who does not, they may have one of the languages CfMC supports for REFORMAT conversions. The types of data definition files supported are as follows:

- CfMC Survent: Questionnaire specifications matching the spread output
- CfMC Mentor: Mentor tabsets matching the spread data
- COSI: Variable definitions export to CfMC companion WINDOWS-based tabulation/printing/charting package
- SPSS: Definitions to load into this data processing package
- SAS: Another popular data processing package
- QUANTUM: A tabulation package
- UNCLE: Another tabulation package
- Triple-S XML: A data standard for various market research applications

The RFT file exported using these options is created with all response list items spread as response codes (except that SAS multi-response CAT questions are spread as 0/1 variables). LOOP variables are spread such that there is a separate iteration for each item answered in code list order (see *{!RFT_UNWIND_LOOPS*).

Note that non-CfMC packages, although supported, follow rules generated by our clients that may or may not match the types of variables you wish to export for those packages. If you wish to automatically create your own definitions, it is often useful to run a program using the MAP file as a basis for further variable creation.

The data definition file(s) created by REFORMAT contain generally include the question text, new location, type of variable, and code list where applicable. In addition to variable definitions, some exports include automatic table-creating specifications (Mentor, QUANTUM).

The variables created for other packages will generate new names to create separate variables in cases where there is a multi-response question or there are questions inside of a loop. For multiple response CAT questions, numbers 01 through 99 are added to the end of the variable name; if there are more than 99 response codes, an ASCII lettering sequence from AA to ZZ is used.

For LOOPS, letters A through Z are added to the end of the variable name.

For multi-response questions within loops, the name extension is 01A for the first name. For example, a question named HOSPITAL that has 10 responses and 5 loop iterations will have its iterations named HOSPITAL01A through HOSPITAL10E.

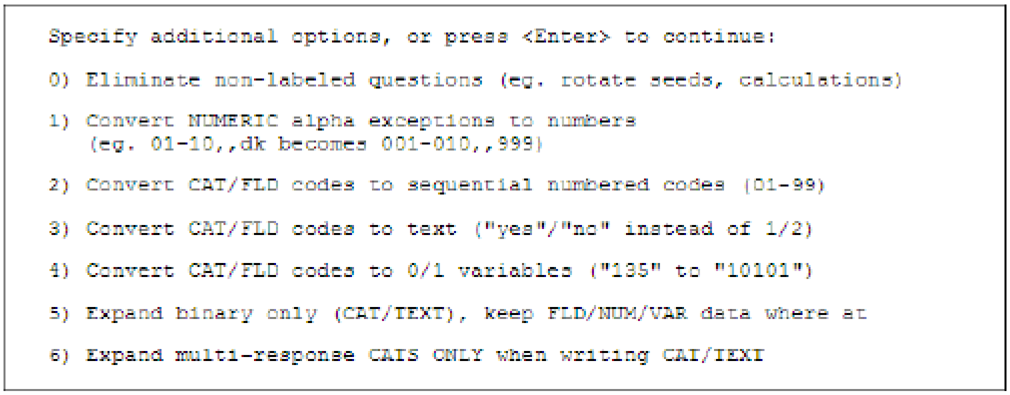
!SPC and !PHONE data types do not get exported to data definition formats. So, if you want to include these you will need to create a !VAR,A or hidden !VAR statements in the location of these statements.

The exported file is saved with a different name extension depending on the package. Survent makes a “qsp” file. For

Mentor or Cosi, it is “def”. For SPSS it is “sps”, SAS is “sas”, QUANTUM is “qua”, UNCLE is “unc” and SSS_XML is “xml”.

Other Data Conversion Options

In addition to the controls you can have in the questionnaire using {!RFT_xxxx} statements, REFORMAT has some other data conversion options. Here is the screen:



```
Specify additional options, or press <Enter> to continue:

0) Eliminate non-labeled questions (eg. rotate seeds, calculations)

1) Convert NUMERIC alpha exceptions to numbers
   (eg. 01-10,,dk becomes 001-010,,999)

2) Convert CAT/FLD codes to sequential numbered codes (01-99)

3) Convert CAT/FLD codes to text ("yes"/"no" instead of 1/2)

4) Convert CAT/FLD codes to 0/1 variables ("135" to "10101")

5) Expand binary only (CAT/TEXT), keep FLD/NUM/VAR data where at

6) Expand multi-response CATS ONLY when writing CAT/TEXT
```

Option 0: This allows you to only convert “named” variables, so you don’t get things used for calculations or other internal purposes.

Option 1: This allows conversion of alphanumeric exceptions on numeric questions to numeric values. The rule applied is that the converted exceptions will be numbered for example 997,998, and 999 and have 1 more digit than the maximum value in the range. So if the range is 1-9, the alphabetic exceptions would be numbered 97,98, and 99. This is for some packages that need numeric values.

Option 2: This changes the code list to be a numbered code list that starts a 1 and goes to “n”, where “n” is the number of codes in the list. This will renumber all codes in the order they are in the code list, and change alphabetic codes to numeric as well.

Option 3: This writes the “Text” of the response into the data instead of the code. It is useful for Excel or database applications where you want to see the printed text instead of the coded value. You must specify the width to use for the code text if you are writing a fixed format file; the maximum is 1000 characters per response and the minimum is 10. This length will apply to ALL coded responses written to the data, so this is best used with a variable length delimited file.

Option 4: Writes multiple response CAT or FLD questions as separate “0/1” variables. That is, one variable is written for each possible answer, with a 1 for yes, and a 0 for no. If this option is chosen, you are prompted as to which types of variables to change, you can choose any of single response !CAT type, multi-response !CAT type (most likely), single response !FLD type, or multi-response !FLD type. Since multi-response !CAT data is binary and unreadable in an ASCII export file, it is often converted this way; the default is for the program to export it the same as multi-response !FLD questions look, that is, with one variable consisting of each response listed after the previous response in the order chosen.

Option 5: This appends CAT question data (in the format specified by the REFORMAT compiler command in your questionnaire file) after all other data. This is useful if you want to leave all other data where it was originally but JUST spread the CAT questions (perhaps because the program you will be processing the data with does not deal with punch data). This also puts the TEXT data at the end of the record.

Option 6: This causes REFORMAT to spread the data for multiple-response CAT questions only. Single response CAT questions are left unchanged from the original data. This option would most likely be used in combination with option 5. For example, if only a few of your CAT questions are multiple-response and you want your reformatted data columns to match the original data file as closely as possible, then use option 6 to spread only the multiple-response CAT questions and option 5 to append them to the end of the spread data file.

Spreading TEX Question as Part of the Standard Data Set

TEXT questions contain long open-end responses. By default the program ignores these when writing out the dataset because of their potential size. If you have a delimited file there is no particular reason to exclude these, but if you are making a fixed format file the size of these can make it hard to use the file so use caution. If you make a CARD IMAGE file the maximum size you can specify is 74 columns (that is all that will fit on one “card”).

If you want to include these in your fixed format dataset, you ust specify the maximum width for a TEXT questions' data (10-5000), and that width will be applied to ALL text questions in the spread output of the file.

After entering the relevant information, you will be presented with the final review screen. Check your options carefully and press **Enter** to process the data, or **S** to save the Mentor specifications for the run to a file before processing the data. The program will run, you will see a list of the questions processed and notes telling you the new files that were created.

Processing the Data

NOTE: The actual ~REFORMAT keywords to control these options for batch mode processing are in *Appendix B: TILDE COMMANDS, ~REFORMAT*, in the *MENTOR* manual, or the *UTILITIES* manual.

If the program runs successfully, it will produce two files <data file name>^RFT and <QFF file name>^RFL. Possible reasons for the program not running would be core size limitations if your files are very large, or a QFF and TR file that do not match. In this case you will get a warning that REFORMAT has found too many responses in the data, meaning it does not match the QFF file.

```
EX:
(WARN #891) There were too many responses found in
the data here! Error on question 0.10 and case ID
0001
```

The program will attempt to write data successfully even if errors are found when matching the data to the variables, so it is *your* responsibility to check the integrity of the data set you create.

The program exits after processing the data file. Within the data records, the original data is spread according to the map file, including any of the REFORMAT program options you may have chosen. The first line of a delimited output RFT file will be a delimited list of the names of the variables written to the file (i.e., ~REFORMAT option DELIMIT_NAME_FIRST is the default.).

5.7.1 Spread Data File Format (RFT) file

The data record has the following format:

The case ID starts in column 1. If using card-image format, a two-column record ID is next. Otherwise the spread data begins in the first column after the case ID (continuing through the actual length of the data) that is interpreted with the RFL file. The ASCII data records can be up to 100,000 columns long. Here is a sample fixed format data record:

```
EX:
00013       7     5     MRS. SALLY FIELDS  101 AZ
```

If using delimited format, the first record will be the variable names in quotes and separated by your delimiter. The next record is the first actual data record, with the delimiter placed between the case ID and each succeeding variable. Quotes are placed around string variables. Here is a sample comma delimited data record with variable names first:

```
          EX:
"case_ident","yrshouse","Numadult","Name","CdtyP01",CdtyP02","State"
"00013",7,5,"MRS. SALLY FIELDS",1,0,"AL"
```

Rules for Spreading Data Variables and Controlling Variables from the Questionnaire

Data from all data-generating questions except TEXT questions is included in the data records by default unless specifically excluded by the !-RFTON command in the questionnaire or because you are getting a subset of the variables from a variables DB file (see *3.2.4 DATA CONTROL COMMANDS, REFORMAT Data Controls And Use*). TEXT questions may be included as an option.

Numeric or string data (from NUM, FLD, VAR, and all other string type questions) is moved directly into the next available data column by default.

In addition to the controls listed above for CATegory and FieLD questions, you can control the spreading of CAT question binary data directly from the questionnaire using the following compiler commands:

{!RFT_CAT_01} This spreads the data as either a code of '1' or '0' to mark the presence or absence of each of the possible codes.

{!RFT_CAT_RESP} The response codes themselves are placed in the data in response order, up to the maximum number of responses (this is the default).

{!RFT_UNWIND_CODES} The response codes are placed in the data, but they are positioned in their relative position from the response list, e.g. if you have the 3rd response, it will be written to the third code location in the variable. Data will look like “bbbb03bb…” where “b” is blank. This will also affect FLD question output.

{!RFT_CAT_PUNCH} Punches 1-9,0,X, and Y for single response CAT questions are placed in the data. The same punches are spread for multipleresponse CAT questions, but in a grid of column sets (number of columns times the maximum responses allowed). Punches are spread in their column set relative to their column position to the start or base column.

{!RFT_CAT_SPREAD} Punches 1-9,0,X, and Y for multiple-response CAT questions are placed in the data. Responses are spread as one punch per column in response list order regardless of the original column order. This is like the default of 1's and 0's except that each column will either have the original punch or be blank.

NOTE: Response list order means the order that responses appear in on the actual response table and *not* the order in which they were given.

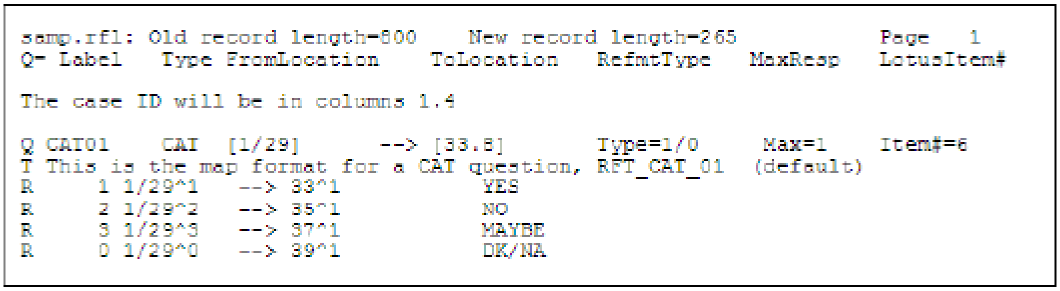
Response codes that contain only carets (^) will appear as carets in the spread data file when responses are spread as codes.

```
EX:
01 YES
02 NO
^^ DK/NA <-- this response will appear as two
carets ^^ in the spread data file.
```

Data from LOOPs is spread according to the controlling response; the third controlling response's data will go into the third data space. Or, you can force the program to place the data sequentially for each response answered using the {!RFT_SAVE_LOOPS} command in the questionnaire.

5.7.2 Map of the RFT File (RFL file)

The map file (RFL extension) lists information about the question data, recode values, questions and responses, and additional information depending on the question type. Here is some sample output using a CAT question formatted with the default CAT reformat options and REFORMAT program Option 2 for LOTUS format:



```
samp.rfl: Old record length=800    New record length=265           Page    1
Q- Label    Type FromLocation     ToLocation    RefmtType    MaxResp    LotusItem#

The case ID will be in columns 1.4

Q CAT01     CAT  [1/29]       --> [33.8]        Type=1/0      Max=1     Item#=6
T This is the map format for a CAT question, RFT_CAT_01  (default)
R      1 1/29^1   --> 33^1           YES
R      2 1/29^2   --> 35^1           NO
R      3 1/29^3   --> 37^1           MAYBE
R      0 1/29^0   --> 39^1           DK/NA
```

The record types printed are identified by the letter in column one:

Letter	**Description**
Q	Question description information
X	Extra line for some CATs, NUM, EXP, and SPC questions
T	Text line
R	Recode item for CAT and FLD questions
L	LOOP data line
D	LOOP data location description line

Q line question descriptions include:

Item	**Description**
Q=label	The question label or number (e.g., CAT01)
Type	Question type (e.g., CAT)
FromLocation	Data location and width in the original data (e.g., [1/29])
ToLocation	Recode item for CAT and FLD questions

NOTE: For TEX questions this will say '--> Not Moved'

Comma-delimited format adds from one to three columns to the original data width (one comma plus "double quotes" around text data).

For CATquestions:

RefmtType	**The Reformat type**
Type=1/0	(default, see {!RFT_CAT_01} or option 7)
Type=CODE	Question type (e.g., CAT)
Type=PUNCH	Data location and width in the original data (e.g., [1/29])
Type=SPREAD	Recode item for CAT and FLD questions
Type=MOVE	(see option 5)

For FLD questions:

RefmtType	**The Reformat type**
Type=MFLD	for multiple-response FLD questions
TYPE=	is not specified for single-response FLD questions

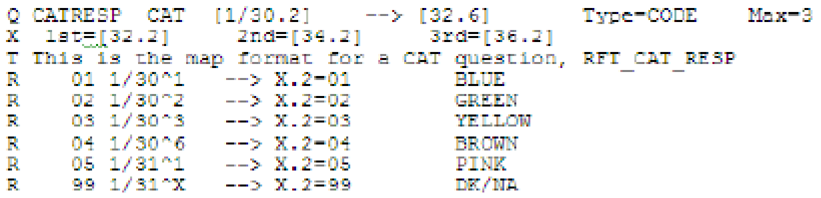
For LOOP questions:

RefmtType	**The Reformat type**
Type=UNWIND	(see {!RFT_UNWIND_LOOPS})
Type=SAVE	(see {!RFT_SAVE_LOOPS})
MaxResp	The number of separate responses allowed (e.g., Max=1)
LotusItem	For comma-delimited format only; this is the Lotus column number for this data (e.g., Item#=6)

X line extra descriptions include:

The column grid that determines the data location in the RFT file for multiple-response CAT questions is spread in either response code or punch format.

Here is an example for a multiple-response CAT question spread as codes:



```
Q CATRESP  CAT  [1/30.2]    --> [32.6]      Type=CODE    Max=3
X  1st=[32.2]     2nd=[34.2]     3rd=[36.2]
T This is the map format for a CAT question, RFT_CAT_RESP
R     01 1/30^1   --> X.2=01          BLUE
R     02 1/30^2   --> X.2=02          GREEN
R     03 1/30^3   --> X.2=03          YELLOW
R     04 1/30^6   --> X.2=04          BROWN
R     05 1/31^1   --> X.2=05          PINK
R     99 1/31^X   --> X.2=99          DK/NA
```

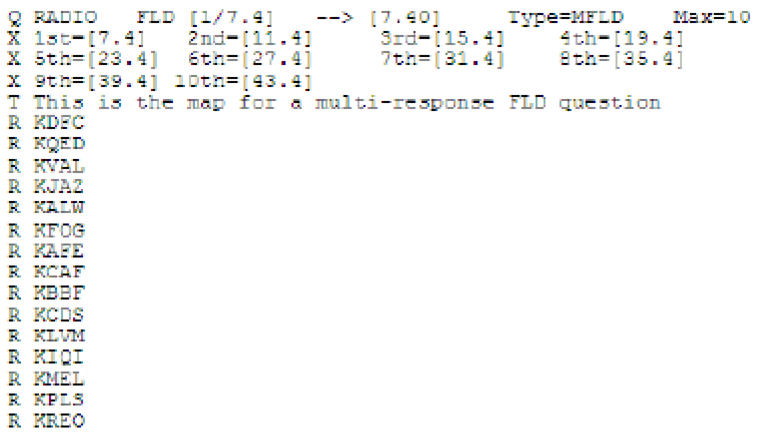
X 1st=[41.2] 2nd=[44.2] 3rd[47.2] where the first two-character response code will be in 41.2, the second in 44.2, and the third in 47.2.

For multiple-response CAT questions spread as punches, the grid of columns indicates where each response will be punched. Responses are spread as punches (one punch per column set) in the order that they appear on the recode table. If the example question was answered 08,04,01,07, then the punch for response 01 would go in the 1st=[41.3], 04 in 2nd=[44.3], 07 in 3rd=[47.3], and 08 in 4th=[50.3]. The position of a punch in its column set is determined by its column position relative to the start of the base column. Here is an example:

```
Q CATPUN2  CAT  [1/35.3]    --> [41.12]      Type=PUNCH   Max=4
X  1st=[41.3]      2nd=[44.3]     3rd=[47.3]      4th=[50.3]
T This is the map format for a CAT question with reformat set to
T RFT_CAT_PUNCH, for a multiple-response question
R     01 1/35^1    --> X+0^1              BLUE
R     02 1/35^2    --> X+0^2              GREEN
R     03 1/36^1    --> X+1^1              YELLOW
R     04 1/37^1    --> X+2^1              BROWN
R     05 1/37^2    --> X+2^2              PINK
R     99 1/35^X    --> X+0^X              DK/NA
```

In this example, if the responses were 01, 03, 05 (or any other order of these same responses) then the punches would be spread as follows: a 1 punch in column 41, a 1 punch in column 45 (column two of the second column set which is offset by one column from the base column of 41), and a 2 punch in column 49 (column three of the third column set which is offset by two columns from the base column of 41). Columns 50, 51 and 52 would be blank since only three of a possible four responses were chosen.

For FLD questions, the X line appears only for multiple-response FLD questions in the same format as multiple-response CAT questions spread as responses ({!RFT_CAT_RESPONSE}). The X line indicates the columns in the RFT file for each of the possible responses. Here is an example:



```
Q RADIO   FLD [1/7.4]   --> [7.40]     Type=MFLD    Max=10
X 1st=[7.4]   2nd=[11.4]     3rd=[15.4]    4th=[19.4]
X 5th=[23.4]  6th=[27.4]     7th=[31.4]    8th=[35.4]
X 9th=[39.4] 10th=[43.4]
T This is the map for a multi-response FLD question
R KDFC
R KQED
R KVAL
R KJAZ
R KALW
R KFOG
R KAFE
R KCAF
R KBBF
R KCDS
R KLVM
R KIQI
R KMEL
R KPLS
R KREO
```

In this example, there are 10 possible responses and each response uses four columns for a total of 40 columns.

For NUM questions, the X line indicates the valid range of responses and any allowed exception codes.

```
EX:
X Range=1-5000          Exceptions=DK,NA
```

For EXP questions, the X line contains the expression for as many lines as are needed.

```
EX:
X (Numqn*100)/2
```

For PHONE questions, the X line indicates the subtype and gives a brief description of that PHONE's subtype.

```
EX:
X SubType=G             Phone Record Information
```

For SPC questions, the X line indicates the subtype and gives a brief description of that SPC's subtype.

```
EX:
X SubType=V  Suspend/Terminate Question Number
```

R line recode descriptions (CAT or FLD) include:

- response code
- for CAT questions, the original data location and punch
- for CAT questions, location and punch or response code in the spread data file
- response text (truncated at column 80 for long lines or multiple lines)

R lines To Location for CAT questions vary for the reformat type chosen:

- For {!RFT_CAT_01}, {!RFT_CAT_SPREAD}, or {!RFT_CAT_PUNCH} for single response you will see the to location and punch in the spread data file.
- For {!RFT_CAT_RESP} and {!RFT_UNWIND_CODES} or option 5 in the REFORMAT program you will see X.width and the response code, where X refers to the spread data column locations on the X line.

```
EX:
X 1st=[32.2] 2nd=[34.2] 3rd=[36.2]
R 03 1/30^3 --> X.2=03 YELLOW
```

In this example, if 03 was the first response for this question, then the code 03 would be in columns 32 and 33 in the spread data file. For {!RFT_CAT_PUNCH} for a multiple-response question you will see X+offset (where X refers to the column is set on X line and offset is the number of columns this column is offset from the base column for this question) and the punch.

```
EX:
X 1st=[41.3] 2nd=[44.3] 3rd=[47.3] 4th=[50.3]
R 03 1/36^1 --> X+1^1 YELLOW
```

where:

X is the base column (in this case 41 in the first column set)

NOTE: The ToLocation field format is determined by whether a single ASCII record or ASCII card images are produced.

T Lines text lines contain:

the text of the question truncated at column 80 for each line of text.

L Line LOOP Data Lines print:

The original data location of the loop controlling response and the corresponding location in the spread data file for the maximum number of times the LOOP was executed. LOOP lines for each question executed inside the LOOP will print the original data location for each LOOP iteration and its corresponding location in the spread data file (see the LOOP example after the D line description).

D Line LOOP Description only prints:

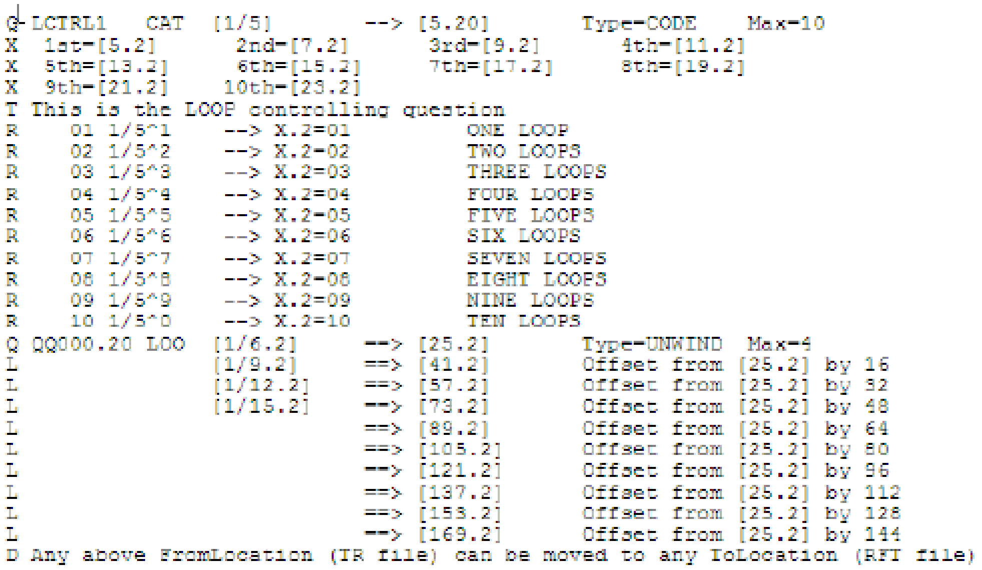
for the {!RFT_UNWIND_LOOPS} compiler option, after the L lines indicating how spread data locations will vary from the original data file (see the LOOP example below).

{!RFT_UNWIND_LOOPS} (the default), puts aside the total number of columns for the LOOP controlling question. LOOP data is spread according to the responses chosen in the controlling question, leaving blank data columns for responses not chosen.

For an UNWIND_LOOPS example, see the following page.

NOTE: With UNWIND_LOOPS, LOOP data is not spread in the order responses were given for the controlling question, but in the order responses appear in the recode table.

Here is an example of UNWIND_LOOPS where the LOOP controlling question allows up to ten responses but the maximum times through the LOOP is four. The data location for each LOOP controlling response in the spread data file is determined by which responses were chosen. For instance, if the LOOP controller question was answered 09, 05, 01, then response 01 would go into columns 25 and 26 (continuing for all the questions executed inside the LOOP), response 05 would start in columns 45 and 46, and response 09 in columns 65 and 66.



```
Q LCTRL1   CAT  [1/5]        --> [5.20]       Type=CODE    Max=10
X  1st=[5.2]       2nd=[7.2]      3rd=[9.2]      4th=[11.2]
X  5th=[13.2]      6th=[15.2]     7th=[17.2]     8th=[19.2]
X  9th=[21.2]     10th=[23.2]
T This is the LOOP controlling question
R    01 1/5^1     --> X.2=01          ONE LOOP
R    02 1/5^2     --> X.2=02          TWO LOOPS
R    03 1/5^3     --> X.2=03          THREE LOOPS
R    04 1/5^4     --> X.2=04          FOUR LOOPS
R    05 1/5^5     --> X.2=05          FIVE LOOPS
R    06 1/5^6     --> X.2=06          SIX LOOPS
R    07 1/5^7     --> X.2=07          SEVEN LOOPS
R    08 1/5^8     --> X.2=08          EIGHT LOOPS
R    09 1/5^9     --> X.2=09          NINE LOOPS
R    10 1/5^0     --> X.2=10          TEN LOOPS
Q QQ000.20 LOO  [1/6.2]      ==> [25.2]       Type=UNWIND  Max=4
L               [1/9.2]      ==> [41.2]       Offset from [25.2] by 16
L               [1/12.2]     ==> [57.2]       Offset from [25.2] by 32
L               [1/15.2]     ==> [73.2]       Offset from [25.2] by 48
L                            ==> [89.2]       Offset from [25.2] by 64
L                            ==> [105.2]      Offset from [25.2] by 80
L                            ==> [121.2]      Offset from [25.2] by 96
L                            ==> [137.2]      Offset from [25.2] by 112
L                            ==> [153.2]      Offset from [25.2] by 128
L                            ==> [169.2]      Offset from [25.2] by 144
D Any above FromLocation (TR file) can be moved to any ToLocation (RFT file)
```

The double-lined arrow (==>) indicates that there is no direct correspondence to the original data locations. Loop data from the original data locations could go into any one of the ten locations listed above depending on which responses from the controlling question were chosen.

Here is the first question executed in the LOOP.

```
Q BRANDS     CAI   [1/8.2]      ==> [27.14]       Type=CODE     Max=7
L                  [1/11.2]     --> [43.14]       Offset from [27.14] by 16
L                  [1/14.2]     --> [59.14]       Offset from [27.14] by 32
L                  [1/17.2]     ==> [75.14]       Offset from [27.14] by 48
L                               --> [91.14]       Offset from [27.14] by 64
L                               ==> [107.14]      Offset from [27.14] by 80
L                               ==> [123.14]      Offset from [27.14] by 96
L                               --> [139.14]      Offset from [27.14] by 112
L                               ==> [155.14]      Offset from [27.14] by 128
L                               ==> [171.14]      Offset from [27.14] by 144
D Any above FromLocation (TR file) can be moved to any ToLocation (RFT file)
X  1st=[27.2]      2nd=[29.2]      3rd=[31.2]      4th=[33.2]
X  5th=[35.2]      6th=[37.2]      7th=[39.2]
T First question inside the LOOP block
R     01 1/8^1     --> X.2=01          COKE
R     02 1/8^2     --> X.2=02          COCA-COLA CLASSIC
R     03 1/8^3     --> X.2=03          NEW COKE
R     04 1/8^4     --> X.2=04          DIET COKE
R     05 1/8^5     --> X.2=05          CHERRY COKE
R     06 1/8^6     --> X.2=06          CAFFEINE FREE COKE
R     98 1/8^7     --> X.2=98          OTHER
R     99 1/8^8     --> X.2=99          DK/NA
```

The actual spread data location for a particular response would be its X line location plus the LOOP offset for the loop iteration in which it was answered. For instance, if 04 was the second response for the BRANDS question for response 08 of the LOOP controlling question, then 04 would be in columns 141 and 142 (the X line location for the first response 29.2 plus the LOOP controller location offset 112).

For {!RFT_SAVE_LOOPS} there is a direct correspondence (indicated by the single --> arrow) between the original data locations for each LOOP iteration and the location in the spread data file. The data would be placed sequentially for each response answered. Using the example above, the LOOP and LOOP questions would show four original data locations and four spread data locations.

NOTE: {!RFT_UNWIND_LOOPS} spreads LOOP data according to the response code order of the controlling question and not in the order the responses were given. The {!RFT_SAVE_LOOPS} option will maintain the original response order so that your spread LOOP data will be in the same order as your TR data file.

~REFORMAT Command Language for Batch Mode operations

You can execute REFORMAT in batch mode by choosing to save the spec file by using the **S** option on the recap screen and then running that saved file through Mentor, or by writing your own command language file to execute REFORMAT commands. Here is a list of the relevant commands and their meanings:

The following commands are required:

~INPUT <filename> <filetype> Name of the data file to process

~QFFFILE <qff file name> Name of the questionnaire file to process

~REFORMAT Invokes the default REFORMAT options

~END Ends the program

The following are ~REFORMAT options, placed after the ~REFORMAT command and before the ~END command:

1 File type options:

DELIMITER=COMMA/TAB/SPACE/<char> Write delimited file with delimiter of <char>

DELIMIT_NAME_FIRST Write name of variable on first line of delimited file

DELIMIT_FIXED_LENGTH Write fixed-length variables instead of the default that strips blanks from the end of variables.

CARD_IMAGE:# Write card image file (80 byte records); by default it writes a case id and 2 character card ID on each record; say :3 to generate a 3-digit card ID.

2 Miscellaneous options:

DO_WHAT_YOU_CAN If there are data errors, attempt to make file anyway

EXPORT_LEVEL=# Used with {!Export_Level=#} in prepare, controls which variables to write. Variables >= the Export Level will be written.

-USE_CFM_LABELS Says to exclude any questions that were not specifically labeled by the user.

3 Map file options:

-MAP_FILE Don't make map file

MAPFILE=(Start list of map file options:

-FROM_FILE_INFO Don't include data locations data came FROM, only TO

-HEADERS Don't include page headings

-FORM_FEED Don't include form feed at page breaks

PAGE_LENGTH=# Change page length from 66 lines per page to #

EXCEL_NAMES Use "AA" - "ZZ" type variable names in delimited files instead of real names

SAMEAS_IN_MAP Write "SAME RESPONSE LIST AS QUESTION xx ABOVE" instead of remaking list if list uses "SAMEAS" feature in Survent.

4 CAT/FLD question options:

APPEND_CAT_DATA Leave other data alone, spread category question data and TEXT data at the end

MULTI_CAT_01 Write 0/1 codes (and matching variables) for multi-response category questions

SINGLE_CAT_01 Write 0/1 codes and variables for single-response category questions

MULTI_FLD_01 Write 0/1 codes and variables for multi-response field questions

SINGLE_FLD_01 Write 0/1 codes and variables for single-response field questions

-EXPAND_SINGLE_CATS Don't expand single response category questions; leave as punches

RENUMBER_RESPONSES Changes the response codes to sequential numbered codes starting at 1 and going to n, where “n” is the number of responses. For instance, if you have a code list with "A/B/C/D/E" codes, they will be converted to "1/2/3/4/5". If you have response codes of "1/2/3/4/5/9" it is converted to "1/2/3/4/5/6". "5/4/3/2/1" codes would be converted to "1/2/3/4/5".

SINGLE_CAT_RESPONSE Expand single response category questions as response codes (override 0/1 coding)

USE_RESPONSE_TEXT=#### Writes the text of the response to the reformat file instead of the code, #### is the maximum length of response and can be 10-1000. This is useful for conversion to databases or spreadsheets where the text is preferable. If not specified, the program will export all the text up to the maximum length of text on the response list. The maximum value is 1000.

If you ask for "mentor" specs, and the longest text on the response list is 72 characters or less, reformat will build a FIELD variable in the exported ^def file, so mentor can create crosstabs with the text, otherwise the data from this will be converted to a VAR (string) type for export to other softwares.

5 NUMERIC question options:

RECODE_ALPHA_EXCEPTIONS Changes alphabetic exception codes to numeric values. For example, the values will be made to be 97,98 and 99 if the range is 1-9.

6 TEX question options:

APPEND_CAT_DATA Leaves other data alone, spread category question data and TEXT data at the end

TEXT_AS_VAR=# Expand TEX questions in standard text area using length of #

DO_TEXT_DATA=# Write “H” header lines and "T" lines for TEX data and wrap at position #

-DO_TEXT_DATA Don't write “H” header lines or "T" lines

TEXT_HOLD_SIZE=# Allow # TEX questions per case instead of 125

7 Data definition file (DEF) options:

SURVENT_SPECS Survent variables matching spread data,

HARDCOPY associated Survent files

QSP

CHK

SUM

MENTOR Mentor table definitions (.def)

CLN Mentor Cleaning specs (.cln)

COSI_SPECS COSI import file (.def)

QUANTUM_SPECS Quantum table definitions (.qua)

SPSS_SPECS SPSS variables (.sps)

SAS_SPECS SAS variables (.sas)

SSS_XML Triple-S XML variables (.xml)

UNCLE_SPECS Uncle table definitions (.unc)

~SPEC_RULES Command Language affecting definition files from REFORMAT

These commands affect the creation of the "definition" files created by reformat:

8 QUANTUM:

ABSOLUTE_QUANTUM_LOCATIONS Puts locations in "absolute" format instead of the default "card/column" format

9 SPSS:

SPSS_LABEL_TRUNCATION Causes labels to be truncated to allow older versions of SPSS to use them.

10 SSS_XML:

ONLY_NUMERIC_EXCEPTIONS Checks to make sure numeric variables do not have alphabetic exceptions

RECODE_ACCEPTABLE_LEVEL=x Gives errors if response lists are not in the allowable format, possible options are:

- Minimum_width_sequential_numeric: must be numeric, with each value 1 greater than the previous value; all entries must be the same width.
- Sequential_numeric: must be numeric and sequential.
- Numeric: must be numeric.
- Alphanumeric: must be alphanumeric (a-z or 0-9). Special_characters: any characters, eg _ ,..

RECODE_WARNING_LEVEL=x Same as above, but it gives you a warning.

5.7.3 Header and Text record using the "Do_Text_Data" option

Although this is not one of the options in the utility menus, you can get a "header" and "text" records listed separately from the data records. This lets you see the text separately from the data, and provides some additional information in the header records. See the Utilities manual for more on the ~reformat commands.

Below is a description of the header record if you ever use the "DO_TEXT_DATA" option.

Header Record Type Layout

The HEADER record has the following format:

Column	Description
1	H
2-11	case ID (left-justified)
12-15	blank
16-19	study ID (first four characters of study name)
20-22	blank
23-26	interviewer ID
30-37	study name (left-justified)
38-39	blank
40-45	flags

46	blank
47-54	case flags

```
EX:
H0001 bank intv bank1 Flags: A
```

Here is a list of possible case flags:

Flag	Description
A	Altered
C	Cleaned
E	Error
F	had webSurvent fast resume
L	Locked
R	Was resumed
S	Sequence error in webSurvent
U	Updated
I	Data created by Random Data Generation (RDG)
r	Resume error
u	Resume update error

Tex Record Type Layout

If spreading TEX data separately, you will only get text records for those cases with TEX question responses (and you have the option of writing only these responses to the spread data file or suppressing them).

The text record has the following format:

Columns	Information
1	T
2-3	QQ
4-9	Question number
10	Blank
11-?	Text (continuing through the actual length of the text, breaking at column 80 or your specified length) to additional lines if needed.

Note: Each additional text line begins with the same format as the first line, i.e. TQQ###.## text. You will get a separate text record for each TEX question answered. The text records come after the data records for a case in the spread data file.

```
EX:
TQQ010.10 Bank of California
```

TEX,B questions with no answer will also generate a record. Starting in column 11 is the string '<blank>'.

```
EX:
TQQ020.00 <blank>
```

Use the TEXT_AS_VAR option to put TEX data into the regular data area instead of this option.

Note: For a similar output, but providing question labels instead of question numbers and in a more readable format, use the LIST utility and ask for "ALL TEXT QUESTIONS".

5.8 MONITORING WITH SURVMON

The SURVMON program can be run on any station on your network.

To run SURVMON, enter:

```
MONITOR
```

The program will list the active studies, the server each is running on and the number of active interviewers on each study. If a particular study had been run on a different server then this information is also provided, but only for problem debugging purposes.

At the program prompt you have a list of choices:

- enter the name of the study you want to monitor
- review the quotas for an active study
- set the pause time for TEX and non-TEX question

- quit

After entering the name of the study you want to monitor (and any password that has been assigned to it), the following information is displayed:

- current time
- study comment
- list of all supervised and unsupervised stations by station number and their corresponding interviewer ID. This can be eliminated by using "SURVMON NO_LIST" or "MONITOR, NOLIST".

The information shown on this screen is updated every 30 seconds, so every time you return to this screen you will get updated information.

In addition, if the questionnaire contains any SPC,E,1 statements the text from the last one executed will display along with the interviewer ID and station number. Enter either the station number (LDEV) or the ID of the interviewer you want to monitor.

Also, you will see the case ID, starts, and completes between interviews. There can be up to five monitors per interviewer.

Restricting Access

The parameter STUDY= on the SURVMON command line restricts the monitor to a particular study. This is used to allow clients to view their own studies but no others. This parameter may be used in conjunction with the NO_LIST parameter to restrict the interviewer list.

To set up a command file for SURVMON using the STUDY= parameter, copy the file MONITOR in the CfMC GO directory (or GO group); and add "STUDY=<varname>" to the SURVMON command line. Then save the file and call it MONSTUDY or such. You would then enter "MONSTUDY MYSTUDY" to monitor the study called MYSTUDY.

```
EX:
SURVMON STUDY=%1                    (DOS)
survmon study=$1 con con            (UNIX)
```

Rating the Monitored Session

END_MON_SURV

The END_MON_SURV causes the program to automatically run a Survent interview at the end of each monitored interview, usually so that the interviewer may be rated by the monitor. The syntax is:

```
     END_MON_SURV <monitorname>
```

where "<monitorname>" must be an interviewer ID in the EMPLOYEE ID file. The program sends the following command:

```
SURVENT con con “INIT:m%s1^qff;%s2;Y;;%s3;%s4;%s5;%s6;%s7;%s8;%s9;”
```

Where "%s1"-"%s9" are filled in as follows:

```
%s1 = monitor name from END_MON_SURV command
%s2 = monitor name from END_MON_SURV command
%s3 = case id of last interview
%s4 = interviewer id monitored
%s5 = device number of monitored interviewer
%s6 = # of starts so far
%s7 = # of completes so far
%s8 = study name being monitored
%s9 = device number of supervisor or monitor
```

The parameters fill in questions in the Survent questionnaire you write for the monitor. This way each monitor will create their own data file with the monitor’s ratings in it. You MUST have a questionnaire for any monitor who wishes to use this facility. Note that two monitors cannot run the same questionnaire from different stations.

Use "-END_MON_SURV" to turn this feature off. "END_MON_SURV" with no ID will say what

the present ID is between colons, e.g. :smcr: or :: if not set. This command also works in the SURVSUPR program.

Here is a simple example questionnaire to do monitor rating:

```
     >purgesame
     ~prep compile -specs
     [msmcr,oneinterview]
```

```
''test monitor quality control
{ lcaseid:
last caseid \_..........
!var,,10 }
{ intvid:
\ainterviewer id \_....
!var,,4 }
{ intvldev:
interviewer ldev \_....
!var,,4 }
{ nstart:
# starts \_.....
!var,n,5 }
{ ncomplt:
# completes
!var,n,5 }
{ studynm:
study name
!var,,10 }
{ suprldev:
supervisor ldev
!var,,10 }
{ evalstuf:
here would be evaluation from ldev \:suprldev: of
\:intvid: on dev \:intvldev
caseid \:lcaseid: who has started \:nstart:
and completed \:ncomplt:
on study \:studynm:
!var }
{
done
!var }
~end
```

Stop Monitoring/Send a Message to Interviewer

Press **Ctrl-Y** (DOS) or **Ctrl-BREAK** (UNIX) to interrupt the SURVMON program. At the prompt you will be able to:

- enter a message to be sent to the interviewer being monitored (and continue monitoring)

- unless you specified NO_LIST as described above, enter ‘Q’ to quit monitoring this station
- press **Enter** to continue monitoring the station (no message sent)

You may send an approximately 80-character message to the interviewer. The message will appear on the interviewer’s screen when the next question displays. ***NOTE:*** After sending a message SURVMON must resynchronize the external monitor and interviewer station; you may notice a delay for approximately two questions before this is accomplished.

Enter ‘Q’ to quit monitoring the station. You will be returned to the main program prompt. From here you can enter the name of another study to monitor, press **Enter** for the current study, display quotas for the current or another study, change the wait time for questions or quit and exit the program.

QSS Enter the command QSS to show the quotas for any of the active studies listed. Quotas cannot be updated from SURVMON. See *4.4.2 SURVSUPR MAIN MENU* for information on modifying quotas for an active study.

WAIT <nonTEX question delay> Before entering the name of the study to monitor, you can set the time to wait before the program clears the screen. Enter the command WAIT followed by a number. This number says how many tenths of seconds to wait before clearing the screen of non-TEX questions. The number can be between 1 and 50. The TEX question delay is set to five times the non-TEX delay.

```
EX:
WAIT 15
```

QUIT Enter QUIT to exit the program

NOTE:

- It is not necessary for the external monitor and the interviewing station being monitored to be the same terminal type (UNIX).
- An interviewer has no knowledge of whether or not they are being monitored.

- Do not monitor a simulated interview which is using the random data generator. This may cause an unrealistic strain on the system.
- A message is sent to the CfMC SERVER and entered into the log file when monitoring either starts or ends, including: date and time, station number of the external monitor, and the number of the monitored station.
- The compiler command -ALLOW_MONITOR will affect what you can monitor.
- The monitor screen displays a message in the bottom right corner indicating who is being monitored. "MON #1234" indicates device #1234 is being monitored, while "MON FRED" says an interviewer with the ID of FRED is being monitored.
- If using an EIS predictive dialer, you can also use the PHONE command. The syntax:

```
PHONE #
```

# is your extension. This gives direct listening to the line of the device being monitored. Both you and the monitored device must be on the dialer.

5.9 MAKECFG: UPDATING CONFIGURATION FILES

The MAKECFG program has two purposes, to rebuild the CfMC server's stations and studies configuration files, and to display the currently active stations and studies.

In UNIX, The stations file is built in the /cfmc/cfg directory (stat761.cfg); ALL study servers on the system look to this file to be sure there are no stations running under multiple servers.

The "studies" file in UNIX is in the local $CFMC/ipcfiles directory for each study server (stdy761.cfg). In DOS, all files are stored together in \CFMC\IPCFILES (DOS) because you cannot have more than one server on the network.

To rebuild the configuration files you say "makecfg makenewfiles" or "makecfg clearmyenv".

The "makenewfiles" option will rebuild the files. The "clearmyenv" option will clear out all entries for a particular server, but will leave entries from other servers in the stations file (UNIX only).

On all platforms you can execute the command CLEARIPC that will run MAKECFG with the "MAKENEWFILES" option.

To run MAKECFG, enter:

```
    MAKECFG
```

Your standard Spec File and List File prompts will appear. You may also enter CON CON or specific file names on the program line. See *2.2 USING PREPARE* if you need more information on these prompts.

Once the List File prompt has been answered, then the following will appear.

Option	Description
Help	Re-displays the initial HELP screen
Show stations	Lists all active stations and servers in the stations file
Show studies	Lists all actice studies in the studies file
Find server	Searches for an active SERVER process. If found, the station number of that SERVER process is returned

QUIT Exits the program.

There are a few additional options not shown on the help screen but that are in the program. You need to enter the password "BAD IDEA" first at the program prompt. Then you can enter "HELP" at the next program prompt. The "BAD IDEA" password is to remind you there are things you can do here that will disable the server or interviewing stations if you execute then improperly while the server is live. You could inadvertently clear the wrong device, study, or the server itself. Please proceed with caution.

CLEAR Clears stations from the server's STATIONS file.

This will not abort current interviews, just clear the station record so the device can be re-started.

Here is the syntax and an example of CLEAR:

```
CLEAR <station(s)>

EX:
CLEAR 60,72
```

The stations can be specified individually or as a range. This can be useful for cleaning up the STATION7 file after a system crash.

```
EX:
CLEAR 50-80
```

CLEAR_STUDY Clears the study from the STUDIES file. This will not clear interviewers off the study. It will just clear the entry so it can be reloaded later. This is used if files are loaded, etc.

Here is the syntax and an example of CLEAR STUDY:

```
CLEARSTUDY <study>

EX:
CLEARSTUDY BK726
```

If the system thinks there's a server there already, you can clear it out by doing the following:

```
FIND SERVER (ldev=n)
BAD IDEA
CLEAR n
```

NOTE: Use all BAD IDEA commands with caution.

MAKE SERVER Causes a server record to be made in case it has somehow gotten lost.

Here is the syntax of MAKE SERVER:

```
MAKE SERVER 25
```

The following are also unlisted, but do not require the use of BAD IDEA first:

DUMP Dumps all information about a device.

```
DUMP <ldev>
```

FINDTTY Displays the information about a device from the TTY file (mailbox, terminal type, time zone, extension).

```
FINDTTY <ldev>
```

SAFE Turns security back on commands after having said BAD IDEA to use CLEAR or MAKE SERVER commands.

SC Shows stations that were previously running in the Stations file but have since quit or have been cleared. This shows the date and time cleared.

SL Shows live stations (same as SHOW STATIONS).

SS Shows study information.

To clear stations under the current CFMCCFG server only (DOS/Unix), you can use the command line parameter CLEARMYENV. Since all stations information is stored in the same STATIONS file, if you have multiple servers running, you may want to clear the entries for one server and not the others.

The CLEARMYENV command operates on the assumption that the current setting of the CFMCCFG variable is the environment that you want cleared, therefore, the CFMCCFG variable MUST be set and must be pointing to the correct directory structure.

Here are examples for UNIX and DOS:

```
setenv CFMCCFG /mycfmc/ipcfiles/ (UNIX)
makecfg CLEARMYENV
SET CfMCCFG=F:\MYCFMC\IPCFILES\ (DOS)
MAKECFG CLEARMYENV
```

WEBS Shows station information for web survent sessions, including the password and process ID.

5.11 FIXRESUM

FIXRESUM is a utility program that reads suspended interview files and a questionnaire file to verify that the suspended interviews can be resumed by the questionnaire. This is used when you have made changes to a questionnaire and wish to be able to restart suspended interviews with the modified questionnaire.

You can test the compatibility of the new version with the old suspend files using FIXRESUM, and if the suspend files will not resume properly, you can note where the inconsistency occurred, and modify the file again to try to make it resumable by Survent.

Using the rules below, you should be able to make changes to questionnaires and resume old suspend files with the new version. If not, you will have to resume them with the old questionnaire version if you wish to save them.

In any case, if you have old suspends to save, be sure to save the old version of the questionnaire (QFF file, QPX file, or QSP file) so that you can resume them in the case where the changes are too drastic to resume them with the newer version.

SURVENT REAL-TIME UPDATING OF SUSPEND FILES

Survent uses the same process as FIXRESUM to update changed suspend files. If the date and time stamp of the questionnaire doesn't match the date/time in the suspended interview it is trying to resume, Survent itself will attempt to update the suspend file, so it is not necessary to run FIXRESUME (see below).

If Survent cannot fix up the file, one of three things will occur depending on the setting of the "FAILEDRESUME: xxxx" option in the server's PARMFILE (see *4.4.4 SERVER PARMFILE PARAMETERS*). By default, it will start the interview over at the beginning. If the FAILEDRESUME option is set to "BLOW", it will blow up and save a data file of the answers so far. If it is set to "STARTHERE", it will go forward as far as it can, then resume the interview at the place where the old suspend file no longer matches the new questionnaire.

This is the preferred option if you want the best chance of resuming old suspended interviews and completing the interviews.

When running FIXRESUM, always back up the resume files you are modifying and save the corresponding questionnaire file. If the FIXRESUM run is partial due to an incomplete fix, you will still be able to resume the complete interview using the old version of the questionnaire and the old suspend file. Once FIXRESUM has been run and does a partial fixup, you cannot resume the new file using the old version. If FIXRESUM should fail completely, you can still resume the suspend files with the old questionnaire.

WHY YOU WOULD USE FIXRESUM

FIXRESUM is useful to check whether the changes you made are resumable with the new version. If the changes you are making will cause many suspends to be lost, you can find that out now rather than when you are live in the field. Also, if you run FIXRESUM and all files are fixed successfully, this speeds up the resume because Survent doesn’t have to attempt to do the FIXRESUM procedure when resuming.

Running the FIXRESUM program

When you run FIXRESUM, it will first prompt you for the new questionnaire file name; you may enter a fully qualified filename, or the study name. If the CFMCQFL variable is set, it will look for the questionnaire file in the standard QFF directory, otherwise it will look in the local directory. Here is an example using a full filename:

```
Type qfile name (or 'quit')--> /usr/cfmc/rel7_6/qff/test1.qff
```

FIXRESUM will next ask whether you want to test the suspend files for compatibility or actually fix them up:

```
We have the study opened; study code: TEST1
Now to adjust suspended interviews to match this version
of the questionnaire
Do you want to T)est fixing or R)eally fix resume files?
```

T

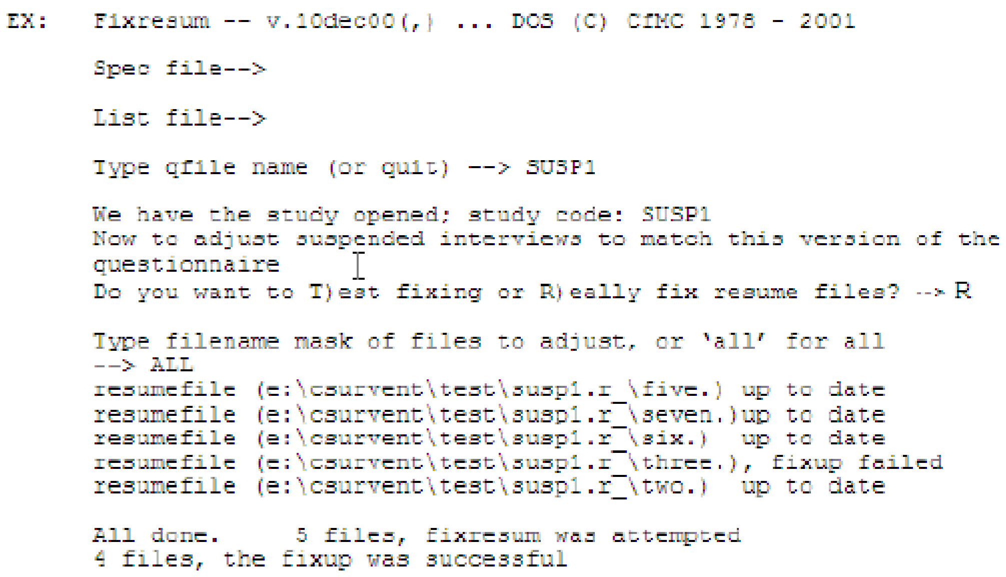
You should choose "T" to test the suspend files and see whether your changes were acceptable. If you type "R" and you are using the "FAILEDRESUME: STARTHERE" option, the program will do a partial fix up of the files it cannot totally fix up and you will not be able to recover the original file to make it work with your changed new version. Once you are satisfied that you have done the best you can to fix up the changes, and you want to go ahead and lose parts of some of the interviews, type "R".

At this point you will be prompted for the files to work on in the resume directory or group. You may enter the keyword "ALL" or a filename mask. The special characters "*" for "ALL, "?" for alphabetic characters, and "#" for numeric filename characters are allowed.

If you have defined the CFMCRESUME variable (DOS or UNIX), FIXRESUM will look in that directory or group for the suspended files. If the CFMCRESUME variable has not been set, FIXRESUM will look in the local subdirectory <study>.R_.

FIXRESUM will read each suspend file along with the new questionnaire and attempt to reconstruct the suspend file so that the responses will match that of the new questionnaire. If FIXRESUM cannot match the suspend file and new questionnaire, it will display the question it was on when it was stopped so that you can possibly fix the problem with that question and recompile the questionnaire to try to get it to match the old suspends. Survent will only fully resume suspended interviews that FIXRESUM can successfully recover.

Here is an example that uses FIXRESUM to update a set of files after testing and changes have been completed:



```
EX:   Fixresum -- v.10dec00(,) ... DOS (C) CfMC 1978 - 2001

      Spec file-->

      List file-->

      Type qfile name (or quit) --> SUSP1

      We have the study opened; study code: SUSP1
      Now to adjust suspended interviews to match this version of the
      questionnaire
      Do you want to T)est fixing or R)eally fix resume files? --> R

      Type filename mask of files to adjust, or 'all' for all
      --> ALL
      resumefile (e:\csurvent\test\susp1.r_\five.) up to date
      resumefile (e:\csurvent\test\susp1.r_\seven.)up to date
      resumefile (e:\csurvent\test\susp1.r_\six.)  up to date
      resumefile (e:\csurvent\test\susp1.r_\three.), fixup failed
      resumefile (e:\csurvent\test\susp1.r_\two.)  up to date

      All done.     5 files, fixresum was attempted
      4 files, the fixup was successful
```

If FIXRESUM is unable to correctly reconstruct a suspend file, and you decide not to resume it with the old questionnaire file version that you saved, you may run the SUSPRES program to print out the responses in the suspend file to a ASCII file (^SUS) and save the data to a CfMC data file (^TR). If you want to save some suspends as if they were completes, you can append them to the data file using the COPYFILE utility or Mentor commands.

Note: You may enter QUIT at the qfile name prompt instead of a questionnaire file name.

Testing Changes

You can test your questionnaire changes to see if your suspend interviews from the previous

version can be resumed with the new version without modifying the suspend files directly.

The prompt says, "Do you want to T)est fixing or R)eally fix RESUM files?" Choose T to test.

5.11.1 FIXRESUM Guidelines

Below is a list of guidelines to maximize the chances that FIXRESUM or Survent will be able to recover all previously suspended interviews.

NOTE: Failure to follow these guidelines may mean that you will not be able to resume interviews suspended prior to your changes!

If possible keep some interviewers using the old version of the questionnaire to complete the interviews suspended prior to the changes. This is the only way to guarantee that all previously suspended interviews can be resumed.

Because this is not usually practical, follow these guidelines for changing a questionnaire already in the field:

1 Back up your originals: Save at least the original questionnaire (QFF), data (TR), phone (FON, FNX), and suspend files in a backup directory or account. You may also want to save the quota (QUO) file (although it is not usually needed), and the spec (QPX) file (although it can be somewhat replicated by using the ~PREPARE MAKE_SPEC_FILES QSP option to create a QSP file). This step is essential to assist in debugging and possibly recovering from any problems. *Never overwrite any of these original files!*

2 Make the necessary modifications to your existing spec file. We strongly recommend that you hardcode all your data locations. Hardcoding data locations may be accomplished by using the HARD_CODE compiler command at the beginning of the spec file and then using the QSP file as your input file for all changes and subsequent runs. If you do not do this you must make sure that any added or deleted questions do not cause the data locations for questions to change. Hardcoding may also be done by using SAVE_COLUMN and RESTORE_COLUMN compiler commands, or by putting all the new questions in the work area (WORK_START=

on the header). FIXRESUM will be unable to reconstruct any suspended interviews containing questions that have had their data locations changed.

NOTE: If you are changing your data locations, you will most likely have larger problem than just the fact that you cannot resume some of your suspend files; data collected in the two different versions of the questionnaire will not correspond.

When making the following modifications to a questionnaire, note these conditions:

- Modifying existing questions: You should be able to change the text or the IF condition on any question without concern. If you change the valid set of answers to any question, you will want to think about how this affects any suspended interviews. For example, you may increase the range on a NUM question without concern, but deleting a response from a CAT question will definitely cause a problem for any suspended interview that originally had that same response.

NOTE: Changing the data position on any question will always cause FIXRESUM to fail to recover any suspended interview where that question was executed.

- Adding new questions: Do not overwrite any data locations with the added questions. Before adding any new questions, use the CHK file to note the highest data column (at the end of the file), and only specify new data locations greater than that column.

- Deleting questions: Delete a question by using one of the following methods:

•

•

- Add the HIDE option on the question line (See *2.3.5 QUESTION LABEL LINE, Question Options* for an example.)

Replace the original question with a GOTO question that does not actually do anything.

· Use HIDE_ALL and -HIDE_ALL compiler commands around the questions to be removed.

Survent re-executes all answered questions when it resumes an interview. By hiding or changing the deleted questions to GOTOs, the program will no longer execute the question, but it will insure that the question will still be found for interviews suspended prior to the changes.

d) Moving questions: If the questions to be moved have no data (DISPLAY, GOTO, etc.) or have data but are labeled, you need to replace the original question with a GOTO question and move the question to wherever it needs to be. If the questions to be moved have data and no label, you may attempt the above procedure, but FIXRESUM will probably not be able to recover those suspended interviews.

The following modifications will cause FIXRESUM to be unsuccessful in recovering suspend files in most cases.

a) Moving the SUSPEND block. It is recommended to always put suspend and resume blocks at the end of a questionnaire and leave them there. Naming the SUSPEND question (i.e., {Suspblock: !SUSPEND}) can help too, so if you add or remove questions before it and then try to resume interviews that were collected before the change, you will eliminate the "Can't find SUSPEND block" message you would otherwise find (if not labeled, Survent uses an internal QQ number that changes every time you compile.)

b) Adding, deleting, or modifying LOOP or ROTATE blocks. If an interview is suspended while in a block of questions where the LOOP or ROTATE was added or modified, the suspended interview cannot be recovered. If a LOOP question is dropped, an interview cannot be recovered if the LOOP question had actually been executed prior to the suspend. If the LOOP or ROTATE question was skipped, the suspended interview can be recovered.

c) Compile and test your new questionnaire. Make certain that your new questionnaire contains the appropriate modifications.

d) Run FIXRESUM on your suspended interview files. If you have made substantial changes to your questionnaire, you may want to test FIXRESUM on a couple of suspended interviews, before running it on all of them. If FIXRESUM is unsuccessful in resuming, try to determine which modifications are causing the problem, and correct them. Then run FIXRESUM on all your suspended interview files.

e) Run SUSPRES on any suspended interviews that FIXRESUM could not fix up. SUSPRES will list out all the responses that were in those suspended interviews.

f) Re-enter the suspended interviews not fixed by FIXRESUM. If you are running without a phone file, just start an interview and enter all the responses from the output from SUSPRES. If you are running with a phone file, you will need to put a PHONE,5 question at the top of the interview to put it in “get specific” mode. Then run FONEUTIL and do the L option with the select of [1087^NB].

This will list out the phone number, suspend record and call back time (if any) of all the suspended interviews. Delete all the suspend records associated with these suspends. Then run Survent and perform a get-specific on the phone numbers of the suspended interviews. You will have to do this twice, because the first time it will give you an error since the suspend record is missing. Be sure to give a callback status so you can get the number again. The second time you get the number, you will want to enter all the responses from the output from SUSPRES and then re-suspend the interview. It is important that you set the call back time correctly if there was one to begin with.

This step is laborious enough that it may be easier to save the old files (questionnaire, phone, and quota), then resume and complete these interviews on the old questionnaire. Start interviewing with your new questionnaire.

5.12 SUSPRES UTILITY

SUSPRES is a utility that allows you to examine files containing suspended interviews. It displays the suspend comment, if any, and the number of the question where the interviewer suspended the interview. You may examine a particular suspend file or all of the files for a study. In addition, SUSPRES produces a file (whose name will be: <suspend file name>.SUS

in DOS and UNIX, containing all of the responses (including the suspend block) for the suspended interviews. This is useful if a suspend file cannot be resumed and the responses must be re-entered, or you just want to see what responses were entered in a particular suspend file. A suspended interview may not be resumable if changes have been made to a questionnaire (i.e., by removing questions), which alter the data locations assigned to the data file in such a way that the FIXRESUM utility cannot match the suspended interview to the new questionnaire. A TR data file will also be produced, which you can add to existing data.

SUSPRES writes all Suspend records to one file, instead of writing each to a separate file. This means users will no longer have problems reading many files as some operating systems (such as DOS) allow reading a limited number of files.

The name of the file will be the same as the study file (eg. bank.tr) but placed within the ${CFMCRESUME}<study>.r_ directory, for instance, /cfmc/resume/bank.r_/bank.tr. Be careful that you remember that this is a file of suspended records, NOT the study data file.

Also, Suspres will report on files that are NOT suspend files instead of trying to read and write them.

When running SUSPRES, it will prompt you for a study code and will then look for the resume directory or group. If you have defined the SET CFMCRESUME variable, SUSPRES will look in that directory (DOS/UNIX) for the suspend files for that study.

Under DOS or UNIX, if the variable SET CFMCRESUME has *not* been defined, then you may only run SUSPRES from the directory which contains the suspend file subdirectory (<studyname>.R_).

After typing the study code you will be prompted to enter either the name of a particular suspend file, ALL to view all suspend files, or QUIT. ***NOTE:*** For filename, wild cards * (all), ? (alpha), # (numeric) are allowed.

```
EX:
```

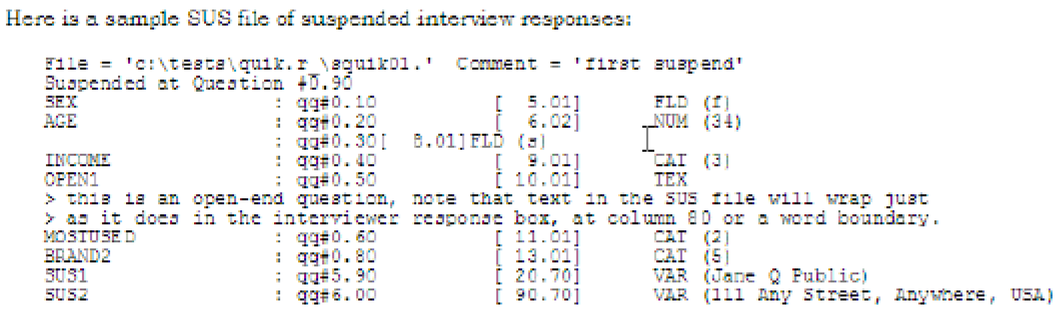
```
Suspres v. 7Dec93(,) ... DOS (C) CfMC 1978 - 1992
Type qfile name-->quik
Enter filename, "all" or Q)uit -->all (or SQUIK* or SQUIK##)
(C:\TESTS\quik.r_\SQUIK01.) comment = 'first suspend'
suspend at question #0.60.......
(C:\TESTS\quik.r_\SQUIK02.) comment = 'second suspend'
suspend at question #0.50.......
```

A SUS file is made for each suspend file that SUSPRES opens. The file contains the suspend file comment and the number of the question where the interview was suspended, followed by information about the questions answered.

```
EX:
Question      Question      Data
label         number        location      Response
SEX           : qq#0.10     [ 5.01]       FLD (f)
```

The actual response to a TEX type question appears immediately after the TEX question. Each line begins with the greater-than “>“ symbol. ***NOTE:*** Unasked TEX questions will have a data location of [0.00]. If the questionnaire contains a SUSPEND block, those responses will appear at the end of the SUS file.

Here is a sample SUS file of suspended interview responses:



Here is a sample SUS file of suspended interview responses:

```
File = 'c:\tests\quik.r_\squik01.'  Comment = 'first suspend'
Suspended at Question #0.90
SEX                  : qq#0.10          [  5.01]      FLD (f)
AGE                  : qq#0.20          [  6.02]      NUM (34)
                     : qq#0.30[  8.01]FLD (s)
INCOME               : qq#0.40          [  9.01]      CAT (3)
OPEN1                : qq#0.50          [ 10.01]      TEX
> this is an open-end question, note that text in the SUS file will wrap just
> as it does in the interviewer response box, at column 80 or a word boundary.
MOSTUSED             : qq#0.60          [ 11.01]      CAT (2)
BRAND2               : qq#0.80          [ 13.01]      CAT (5)
SUS1                 : qq#5.90          [ 20.70]      VAR (Jane Q Public)
SUS2                 : qq#6.00          [ 90.70]      VAR (111 Any Street, Anywhere, USA)
```

5.13 MAKEZONE PROGRAM

This program will let you add area codes to an existing zone table file. Since area codes are being added so quickly, users cannot wait for each update to receive a new file. You can add area codes whenever you want. Be careful about adding area codes into the table before the area is activated, or your interviewers will be unable to successfully dial those numbers.

NOTE to predictive/preview/auto dialer users: Make sure that the dialer program has the same area codes added to its database or the dialer may reject these numbers even though CfMC thinks they're okay.

MAKEZONE will read a spec file that is basically an ASCII version of the zone table. This spec file is the zone source file and is supplied as part of each update. The exact name of the file is operating machine dependent.

```
\CFMC\SURVENT\ZONETABL.SRC                  (DOS)
/cfmc/survent/zonetabl.src                  (UNIX)
```

To add a new area code into the ZONETABL do the following:

3 Make backup copies of the current ZONETABL and zone source files.

4 Try dialing a couple of the numbers that your current zone table is rejecting and verify that the phone rings. If it does ring, then the area code has been activated. If it doesn't ring and you get a fast busy signal, then the area code has not been activated and you probably should not add that area code into the zone table.

5 Check the current zone source file to see if the area code exists. If it does not exist, find the area code it split off from, and if it has no exceptions, you can copy that line to the appropriate place for the new area code and change the area code to the new area code.

The current zone source has been updated with comments about area codes that are not currently activated, but are planned to be activated in the near future. Many of these new area codes have test numbers which you can use to see if the area code has been activated. The source file will also have comments in it for when the file was last updated.

If the old area code has exceptions, you *cannot* update this area code with 100% accuracy. Contact the Support Hotline or check the CfMC bulletin board for information about this area code. CfMC may not have 100% accurate information for some time after the area code is activated. We must rely on our supplier to give us a list of exchanges for the new area code. If you need 100% accuracy and CfMC has not yet received updated information, then your only recourse is to use a Mentor procedure that converts the new area code to the old one. See the file FIXAC^SPX in the Survent directory/group for an example of how to do this.

6 After adding the area code(s) into the zone source file, run MAKEZONE and at the Spec File prompt put in the name of the zone source file. At the List File prompt put in a valid file name to save any error messages. If you get error messages, try to figure out what you did wrong. Most likely you did not supply a proper time zone or daylight savings setting for the area code you added.

7 If MAKEZONE runs successfully, it will create a file called ZONETABL in the local directory/group. You can then use the Mentor spec file ZONE2DOC to convert the Zonetabl back to the source file. Compare the new ZONEDOC file with the zone source file to make sure it contains all your changes.

8 If ZONEDOC has all your proper changes, you can rename the ZONETABL file into the CONTROL directory/group and rename the ZONEDOC file into the Survent directory/group.

9 Use FONEBULD to add some numbers into a dummy phone file. Make sure you use a cross-section of numbers that contain both new and old area codes, plus some known bad area codes like 999. If this works, everything is okay. If it doesn't, retrace your steps to see what went wrong.

NOTE: If you are unable to figure out what is wrong, then use your backup of the ZONETABL and contact the Support Hotline for assistance.

THE PHONE SYSTEM

6

6.1 INTRODUCTION TO PHONE SYSTEM

The Survent phone system is designed to reduce much of the labor and paperwork associated with managing phone numbers for telephone interviewing. The phone system performs the following:

- If a predictive dialer is being used, it assigns the interviewer to a study; the dialer calls numbers until it reaches a person, at which time the call is directed to the available interviewer to complete the survey.
- If a preview dialer is being used, numbers are called for that interviewer until a person is on the phone, at which time the call is directed to the interviewer. In this case the interviewer waits until one of the numbers dialed for them reaches a person with whom to complete a survey.
- If a modem or autodialer is used, the number is dialed by the system, the interviewer waits for the call status and records it or completes the survey.
- If no dialing equipment is directly connected to the computer, it gets the next number for the interviewer; the interviewer dials the numbers and records its status until they reach someone to complete a survey.
- Alternately, the phone system can get specific records requested by phone number, name, address or other information in the sample file (e.g. for respondent call-in studies or Web surveys).
- Once retrieved, the program records the outcome of the call (busy, no answer, etc.) along with any notes associated with the call. If the call is partially completed, it saves the responses and makes them available when the call is continued.
- Unanswered or busy phone calls are rescheduled according to rules that you set up. (You may change the rules at any time.)

Numbers may also be assigned to specific interviewers or interviewer types for special handling.

- Unanswered or busy phone numbers are retrieved according to your rules, keeping a log of each attempt. This feature eliminates the need for comp lex callback schedules.
- Allows interviewers to enter a specific callback time, date, or time-of-day and retrieves the phone number at that time.
- Generates summary reports, including interviewer productivity and call scheduling status reports.

There are many features available in the phone system. Some things are best explained by example. So, throughout the chapter, there are "example file" references. These files are either questionnaire specification files (e.g. PHONE^QPX) or Mentor specification files (e.g. MOVER^SPX) that are working examples of phone system features. (***NOTE:*** for DOS users, Mentor spec files may require a key to be run). The files are located in the /cfmc/survent directory. We encourage you to compile and run these examples to get a clear picture of how the related feature works.

At the heart of the phone system is the phone file. The phone file consists of two parts: the phone parameters and the phone records.

The phone parameters are the set of rules used to dial numbers, such as how many time zones, times available to call, and number of calls to make. There is a default parameter list you maintain that is loaded when the phone file is built.

The phone sample is generally provided by an outside source, or you can generate a sample of phone numbers with Mentor. If provided, it generally comes with information about the business or household at that number. You must put this file in a specific format (telephone # first, text starting in column 51, etc.) using Mentor or some other file processing program (see example file MOVER^SPX and 6.1.1 *Building the Phone File).*

The program FONEBULD reads the formatted sample file to build the phone file for your study. It first reads the standard calling parameters into the phone file (which you may modify as you see fit), then it reads the file of phone numbers. The phone numbers are linked in “stacks” according to your calling rules as the file is built.

Each record in the phone file contains the phone number, user-supplied text, system-related information and a history of each previous call. Survent chooses a phone number to retrieve based on the calling parameter rules and loads it in memory to be used during the interview. It writes back into it data you specifically place there and additional information about what happened to the number (it was dialed but not answered, answered but not qualified, etc.).

To use the phone file for dialing, PHONE statements must be added to the questionnaire. These statements cause the number to be retrieved, display phone information, change parameters, and get or put information to/from the phone record. If the call does not result in a completed interview, the interviewer is prompted to enter a call status or the program provides one; otherwise, the interview is completed and the phone record is marked as having been completed.

When the number receives a non-complete status, Survent reschedules the number according to your calling rules. For example, if the number is busy, the system might schedule it for callback in X minutes. If it is disconnected or inoperative, it will never appear again. Or, it may be scheduled for a callback at a later time that is convenient for the respondent.

Most parameters may be changed by the supervisory program SURVSUPR while the study is in progress (see the MPF option in 4.4.2 SURVSUPR MAIN MENU, Options). Using SURVSUPR, you also have access to information about the numbers: how many have been called, how many are to be called back, the number of disconnects and unused numbers, etc.

In terminal mode, if Survents lose contact with the CfMC server (UNIX/MPE) the phone record will be saved back to the phone file with status 190. In addition, the data record is suspended and saved, allowing you to resume the interview immediately or later on. The call is re-scheduled as a timed call 24 hours later. This is done by checking for the terminal's "SIGHUP" signal periodically; if it disappears the program suspends the call.

FONEUTIL provides many functions regarding management of the phone file. It can report on subsets of the numbers, clear unwanted call histories, verify and/or fix a file, print numbers according to your criteria, or make a copy of the phone records which can be changed and added back in later for callback.

See 6.7.1 *Managing the Phone File with FONEUTIL* for more information.

Mentor and Utilities will read a phone file for reporting, even while the study is active.

6.1.1 Building the Phone File (FONEBULD)

FONEBULD builds the phone file by reading the default "shop" file of calling parameters (or some other file you specify), then reading the sample records provided and loading them in the appropriate “stack” to be called.

The raw file to be loaded into FONEBULD must have the following column-specific format. If necessary, you must use your data processing package or example file MOVER^SPX to get your data into the proper column positions:

Column		
1-9	Phone number	(area code, exchange, number), left-justified. Spaces, parentheses, dashes and other non-digit characters are cleaned to return the remaining digits as the phone number. This is the only REQUIRED field. Note: Phone numbers must be at least 5 digits long.
6-20	ID:xxxx	If assigning interviewer ownership, this specifies the ID of the interviewer to make the call (see 6.6.5 *Ownership Mode*). Note: Notice that this overlaps the phone number, so the phone number is limited to 12 or fewer digits if this option is used. Optionally, you may use the keyword OWNER_LOC=<location.width> and place the ID of the owner there.

Column		
21	not used	Not used (Used by CfMC when reading "ASCIIfied" phone files)
22	Special interviewer type (0-9)	This can be used to assign the number to a special-language interviewer or for some other reason assigned to a particular group of interviewers; the number will go to the standard calling stacks but be moved to the special interviewer stack at the time it is scheduled to be called.
23-24	Time zone (01-24)	This is REQUIRED for time zones or area codes not supported by the CfMC zone table; it is optional if you are using standard North American phone numbers; time zones start at 01 for Greenwich Mean Time; "Eastern time" is time zone 05 and "Pacific time" is time zone 08.
25-34	Case ID (4-10 digits)	If pre-assigning a data record to use with the phone number, this is the data record's case ID (See 6.6.4 *Phone Says Datarecord Mode*). Another option that you can use is CASEID_LOC=<location.width> where you should place the case ID other than in the user text area.

Column		
31-46	Initial calling state	If blank, the number is placed in a 'fresh number' stack. Other options are: 1. The date/time to make the first call. You may use any of the standard CfMC date/time formats, including the format "YYYYMMDDHHMM", where YYYY=year, MM=month, DD=day, HH=hour of day, and MM=minute. For example, "200012231530" would make the first call to the number on DECEMBER 23, 2000 at 3:30pm. 2. "SPECIAL=#" to put the number directly into a special interviewer stack (#=0-9, see also column 22 above). 3. BUCKET=# Where # is 1-8 to call in a particular daypart, or BUCKET=9 to save the number to be called later (see the USE_BUCKET_9 parameter for more information). 4. STACK=# to put the record in a particular stack (see 6.4.1 *How the Phone System Schedules Calls*).
47-49	"Replicate" (001-999)	If this is used, phone numbers are loaded in numbered groups so you can control when the group is released for calling using SURVSUPR. If replicates are used, the sample file must be loaded in replicate order.
50	Not used	(Used by CfMC when reading ASCIIfied phone files)
51-4950	User-supplied information	Information, such as respondent name and address. This can be limited using the TEXTLENGTH=#### command for smaller files.

EX:

```
         1         2         3         4         5         6
123456789012345678901234567890123456789012345678901234567890123456789
2125551212                                                R. Cramden Bus Co.
```

FONEBULD will check for valid phone numbers before loading them. If the phone records are in North America, the program will attempt to verify that the area code and exchange are valid and assigns numbers to the proper time zone. It will also check for duplicate numbers and valid phone parameters if they are preset on a record. Additionally, FONEBULD removes all non-digit characters from phone numbers before trying to process them. This means you can get phone number info in any format (eg. "(415) 444-0470"), and it will be readable into FONEBULD without additional cleaning of the data.

You can create a phone file of up to 999,999 numbers. If you have a limited disk space or a busy system for outbound calling, CfMC recommends that you only load enough numbers to call over a limited time period, because the processes you need to run on the file may take a long time to process for larger files.

In addition, FONEBULD allows you to specify a filename instead of a study code. In particular, this allows you to reference files in other directories when trying to add sample. Fonebuld allows the use of paths or the use of environment variables (for example, !CFMCFONE!) as part of the filename.

FONEBULD supports THE "FONESTAT" file to use different status labels for different studies Fonebuld creates a ".fst" file with the labels to use for this study when you create a .fon file. You can edit this file and copy it to $CFMC/control and Survent, and the phone reports will use the labels as you describe them. By default, the server and phone reports will look for <study>.fst before using the default "fonestat" file in $CFMC/control.

You can also specify a name of a file to use for your study. The file with the name you specify MUST be placed in the $CFMC/control directory. If you use the name "<study>.fst", the phone reports will also pick up the statuses from that file.

FONEBULD supports quota names in phone record so server suppresses overquota records in lieu of market controls

FONEBULD now allows you to specify the location of the quota name you want checked before the server releases the phone number. If the quota name is greater than it's target, the record will be assigned the status you want. This allows you to suppress the release of numbers to a dialer or interviewers that are over quota.

This feature is similar to the "market" feature, but it has a disadvantage because it has to check each record for its quota information and then give statuses based on whether it is over quota or not. Markets will disallow access to the stack if the market weight is set to zero so the records are not processed at all.

The advantage of this feature is that it can be used with as many quota names as you want (up to 1000 triple quotas and even more numbered quotas). Also, used in conjunction with the quota_percentage feature, you can control the release of sample if you have for instance daily quotas. You may also use this in addition to market controls.

The keywords and their meanings are as follows:

Keyword	**Meaning**
quota_name_location=#,#	The location of the quota name in the phone record. The quota name can be up to 14 characters. If the name is all digits it will use a numbered quota.
quota_target_name_location=#.#	The location of the target quota in the record (otherwise it uses quotaname.T if you are using "triple_quotas").
quota_full_phone_status=#	The status to assign if the quota is full
quota_full_callback_minutes=#	The number of minutes to call overquota numbers back in if the status is a callback status.

The server counts the number of records released so far in that quota group and will never release more numbers than what would fill the target quota if all the released records were completes. If the quota is a numbered quota, the server will not keep track of the number of records submitted for each number, so it will be a simple shutdown of the quota once it is full.

Once the quota name value is greater than or equal to the target value, the record is given the "quota_full_phone_status". If no status is specified it gives it status 80 by default. You can specify the status to assign in the parmfile as well using "QUOTA_FULL_PHONE_STATUS: ###", but if you assign a status in FONEBULD that will be the one it uses.

If there is no quota name in the field the record is treated normally. To show the active quotas, say "show_phone_quotas <study>" in the supervisor.

FONEBULD supports quota percents in phone record so server suppresses overquota records in lieu of market controls

Related to the QUOTA_NAME_LOCATION feature, this allows you to specify a percentage of the target that you want to call for now, and give a status for the server to assign if that percentage is met. Then, you can change the percentage to release those numbers later. This way, you are, in effect, spreading calls evenly across the supported quotas as they are filled. You use a quota to set the percentage you want to try currently.

The keywords are:

Keyword	Meaning
quota_percentage_target_location=##	The place in the record where the name of the quota with the target percentage is. This could be updated by a supervisor during the study. If it contains a number it is a numbered quota.
quota_percentage_full_phone_status=#	The status to assign numbers that are attempted to be picked up when their quota is greater than the percentage of the target specified above. This should be an active status > 100 so the numbers can be re-released later.

In addition, FONEBULD supports clusters in phone record

You can now specify phone records in "clusters" such that the server will only hand out one number in that cluster at a time. Once the number is put back in the phone file, it can retrieve another number in that cluster. If you want to control what happens when you get a complete in that cluster you need to do that with quotas or other logic.

The keywords are:

Keyword	Meaning
cluster_location=#.#	Contains a number which identifies the 'cluster' the record is in.
cluster_active_callback_minutes=#	Specifies the number of minutes to schedule numbers rejected because their cluster was busy for. The default is one minute.

To show the cluster activity in the supervisor, type "show_phone_clusters <study>".

The syntax for running FONEBULD is:

```
FONEBULD <listfile>
```

The listed output will go both to the screen and to the disk file named <listfile>.

To read in a file (or part of a file) of commands to begin FONEBULD, this is the syntax:

```
FONEBULD “&myfile.spx(1/20)” con
```

FONEBULD expects an option, HELP for a display of the options, or GO to begin entering file and study names. You can give commands from a file, but it is often used interactively to set initial parameters, etc. Here is the list of FONEBULD options displayed when you type “Help”:

ASCII - Reload an ASCII sample file converted from a CfMC converted sample file using FONEUTIL.

When using the ASCII option to load numbers that have previously been called, if any ascii record has more history slots than the current configured maximum, the program will stop loading numbers with an immediate fatal error UNLESS the FONEBULD command DROP_EXTRA_HISTORY_SLOTS has been specified. If this is the case for ALL ascii records loaded, the first slot and the last n-2 slots will be kept, where n is the number of configured history slots, and all other call histories will be discarded.

COMPRESS - Takes the existing .fon file and removes the space used by the deleted records.

Duplicates_file <name> - Name of file to save the list of numbers that you attempted to add to a file with duplicates=no for some index, such that the number was resolved with status 91 and not available for calling.

DO_NOT_CONTACT file <name> - Name of CfMC system (.tr) file that contains the list of number to compare to when loading records such that if there is a match the record will be put in the "Do not call" stack and will not be available for calling.

Rejects_to_file <name> - Save rejected numbers to file <name>

Stop_after=### - Stop reading phone records at record number ###

Shop_file - Work with phone parameters in interactive mode

Read_shop_file <name> - Read a named shopfile, or 'shopfile' if no <name>

Write_shop_file <name> - Write a shopfile of the current parameters. Will be written with <name> or 'shopfile'

Index loc,wid/no_index - Make (default-fnx)/don't make an index to call specific phone numbers, loc,wid is position for up to 2 additional indices (fny,fnz)

Market_loc <loc,wid> -Location in phone text where market names are read from. The maximum width is 20.

Max_markets <num> - Maximum number of markets allowed

Market <name> [,<number>] [,<weight>] - Name, number, and initial weight for market. Specify all markets in market location or use '???' as a "catchall" category for non-matches.

Header <name><,AFTERGO> - Write phone parameters to an ASCII file now or after “GO”.

<name> - Read an ASCII file of phone parameters

Allowdups=yes/no/### - Whether to allow duplicate phone numbers or index values. There are three values, so use YNN, for instance, to allow duplicates in the phone number field but not other indices, where "Y" is yes and "N" is no.

Textlength <length> - Specify phone text length for this (new) file

Timezones #-#, #, ... - Specify the valid time zones this (new) file

Go - Last command; prompts for study code and file of new/ASCII records to read in and indexes the file

These are most of the FONEBULD commands that do not have anything to do with setting parameters (see *6.2.1 The Phone File Parameters*).

PHONE SYSTEM COMMANDS

Here is a more detailed description of the commands:

ASCII causes the program to expect a file of converted ASCII phone records from FONEUTIL to be read. These records will have system information and call histories that will be saved into the new file. You might use this to move records from one study to another or to rebuild records. The program will only allow these type of records as input if this option is specified, although you can add as many ASCII files as you want in one FONEBULD run.

The ASCII command also allows loading numbers with extra history slots from other projects or suppliers. When using this command to load numbers that have previously been called, if any ascii record has more history slots than the current configured maximum, the program will stop loading numbers with an immediate fatal error UNLESS the FONEBULD command DROP_EXTRA_HISTORY_SLOTS has been specified, in which case for ALL ASCII records loaded, the first slot and the last n-2 slots will be kept, where n is the number of configured history slots, and all other call histories will be discarded.

COMPRESS takes the existing .fon file and removes the space used by the deleted records. This allows you to clean up the empty space in a file without converting the file to ASCII and re-building it. Just say "compress", then "go", then the study code, and "yes" to add numbers/rebuild index.

DO_NOT_CONTACT_FILE <name> is a file that contains numbers that are NEVER allowed to be added to a phone file. This list is used to make it possible (and relatively easy and inexpensive) to honor peoples' requests to not be called by telephone research marketers. The DNC file should be a TR file. Currently, there is no data in the cases. It is an indexed CfMC TR file with phone numbers assigned as the case id for the file. For more detailed information, see section *6.6.2 The Do Not Contact (DNC) File*.

DUPLICATES_FILE <name> is a file of numbers that are allowed to be duplicates even if you have set "allow_duplicates=nnn" (note: full command is "ALLOW_DUPLICATES_FILE").

REJECTS_TO_FILE <name> causes FONEBULD to save "bad" records to file <name>. "Bad" records have badly formatted phone numbers or bad information in one of the other system information fields.

In addition, arguments can be added to REJECTS_TO_FILE command:

```
Syntax:
Rejects_to_file [,-error][,-append]
```

Or, it can be "none" to have output go to the screen only.

The "-error" keyword controls whether the fonebuld run counts rejects as errors or not. The default is, if any input records are rejected then the programs ends with the error flag set. "-Error" means records can be rejected but fonebuld does not end in error.

"Append" means that rejected records will be appended to the specified file. The defaults are that fonebuld ends in error if rejects are found and records are not appended.

There is also the parmfile keyword REJECTFILE: [-error] [append] which can control the defaults.

STOP_AFTER=### causes FONEBULD to only read ### records from the raw file into the phone file. This is often used to build a test phone file.

SHOPFILE brings up the current phone parameters in interactive mode, which can then be modified on the screen. This is the same screen as the MPF screen in SURVSUPR and the FONEUTIL "M" screen. The difference is that there are some parameters that can only be modified here before the phone file is built, such as the number of time zones or phone number size. This command can be used multiple times in a FONEBULD session.

READ_SHOPFILE <name> reads a shopfile previous written to a file with the WRITESHOPFILE command. This will reset all phone parameters to the parameters in the file being read. READ_SHOPFILE NONE may be specified to INITIALIZE the shopfile so you may make parameter changes without having loaded an actual shopfile.

WRITE_SHOPFILE <name> writes the current phone parameters to a file, which can be later be read. To set up your standard ‘shopfile’, run FONEBULD, use the SHOPFILE command, make the changes you want, then save the file to the CfMC CONTROL area with the name SHOPFILE; eg. In DOS, say “WRITESHOPFILE \cfmc\control\shopfile”. You may also make standard files for particular types of studies.

INDEX <loc.wid>/NOINDEX these commands control whether to make indices for the phone file of the phone number and/or other fields. Indices allow the program to retrieve numbers when you ask for a specific number (see the !PHONE,5 statement). NOINDEX says don’t make any index at all. The default is to make an index of the phone number only. INDEX <loc.wid> says to make an index using the field <loc.wid> in the user text. The maximum index width is 14. You can have up to 2 indices in addition to the phone number field. Example: INDEX 51.14 will make an index of the data in the field from column 51-64. The indices are checked for duplication if you have the ALLOW_DUPLICATES option set to NO for that index, e.g., “Allowdups=NNN”.

MARKET_LOC <loc.wid> tells the program where the market name is in the user text area of the phone record (See discussion of MARKETS below). MARKET_NAME_LOCATION 61.8 would say that the market name is in the field 61-68. This may also be filled in on the SHOPFILE screen. The maximum width is 8.

MAX_MARKETS ### tells the program how many markets to allow for. Set this high if you expect to add more markets later (MAXIMUM_MARKETS).

MARKET <name>,<number>,<weight> must be specified for each market that you plan to add. The market number and weight are not required, just the name. If the number is omitted, the next number is assigned. It is a good idea to keep the market name specification in a file to remake the markets later and document the market number/order assigned.

HEADER <name> writes an ASCII list of the phone parameters to file <name>. This list can be edited and read in using "&<name>" in future FONEBULD runs, or just saved for documentation purposes. If you want it to write out the phone parameters after the raw records are compiled, use the syntax "HEADER <name>,AFTERGO" (note: the full command is "ASCII_FONE_HEADER").

&<name> reads in lines from the file <name>, usually previously created with a HEADER command. This might also be a set of MARKET commands or any other FONEBULD command (See the utilities manual for more &<name> options).

GO Causes the program to ask for a study code and the name of a file or files to process, then start processing phone numbers. Sample records are checked for valid syntax and valid phone numbers. If ALLOWDUPS=no, duplicates are resolved with status 91. All valid numbers are added to the phone file and placed in the proper calling stack. If a phone file exists with the study code specify, the program will ask if you wish to add numbers to the existing file, and

will add the new numbers to the file if you say so. The INDEX files are refreshed each time GO is specified as well (See INDEX below).

The rest of the options on the help screen are phone parameters that are discussed in *6.2.1 The Phone File Parameters* below. The following command is not on the help screen but may be needed on occasion:

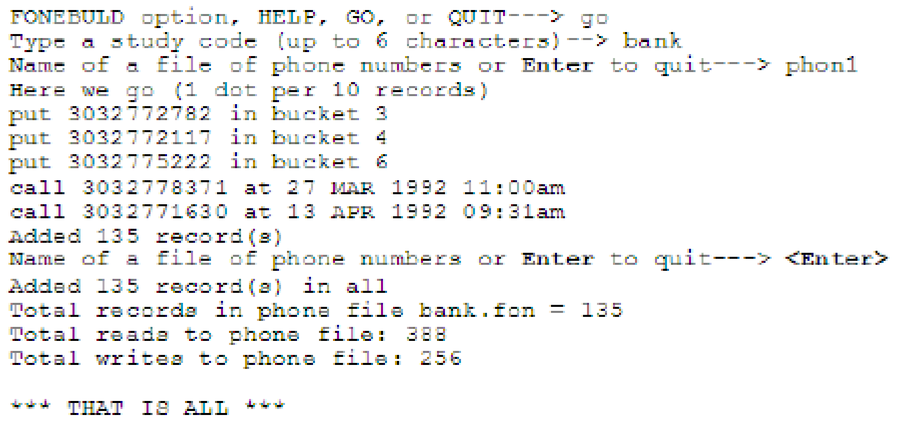
INDEX_FILE_SIZE=###### tells the program the maximum number of items to allow in the indices; the default is twice as many entries as the number of records initially built into the phone file. This is used in case you are adding phone records "on the fly" during the interview and you want to allow for many records to be added later. The current maximum is 999,999.

When building the phone file, if the SHOPFILE option is not used, the initial phone parameters are read from the local file called

SHOPFILE; if there is none, the program uses the shop file in \CFMC\CONTROL.

Here is a listing from a sample run showing the creation of a phone file with FONEBULD:

```
EX:
```



```
FONEBULD option, HELP, GO, or QUIT---> go
Type a study code (up to 6 characters)--> bank
Name of a file of phone numbers or Enter to quit---> phon1
Here we go (1 dot per 10 records)
put 3032772782 in bucket 3
put 3032772117 in bucket 4
put 3032775222 in bucket 6
call 3032778371 at 27 MAR 1992 11:00am
call 3032771630 at 13 APR 1992 09:31am
Added 135 record(s)
Name of a file of phone numbers or Enter to quit---> <Enter>
Added 135 record(s) in all
Total records in phone file bank.fon = 135
Total reads to phone file: 388
Total writes to phone file: 256

*** THAT IS ALL ***
```

In this example, FONEBULD created a phone file called BANK.FON from a file of phone numbers (PHON1), using the standard shop file parameters. If you already have a phone file corresponding to the study code specified, then FONEBULD will ask you if you want to add to the old phone file or if you used an existing name by mistake. You answer the same prompts, but you will see new information as the program finds and reads the existing file and adds to it. You cannot add numbers to an active file with FONEBULD (see 6.3.1 The PHONE Statement, Phone, Letter Subtypes to add numbers to a live file with Survent).

FONEBULD creates the phone file with the parameters taken from the standard shop file (SHOPFILE in the current directory or the CfMC CONTROL area), or the one specified with the SHOPFILE option. The phone file will be named with the study name and the extension FON, e.g., JB123.FON. If INDICES are created for the phone number or other fields, the files <study>.FNX (phone number index file), <study>.FNY (second index file), and <study>.FNZ (third index file) will be created.

FONEBULD looks in two places to determine the time zone of each number. First, it looks in columns 23-24 of the raw file to see if you have told it what time zone it is in. If you have a value there, it uses that. You MUST have a value there for numbers not in NORTH AMERICA. If you do not have a value there and the INTERNATIONAL_DIALING_OPTION is not set, it uses the file \CFMC\CONTROL\ZONETABL, to check the area codes and exchanges and assigns them to the proper time zone in North America. If it does not find a valid time zone match by then, FONEBULD prints an error message, and does not add the number to the phone file.

NOTE: The phone file is by default created indexed. This allows the interviewer to ask for specific phone numbers. See 6.3.1 The Phone Statement for more details on this. An FNX file will be created in addition to the FON file, and both files must be moved to the CFMCFONE subdirectory.

The BREAK key is disabled when adding to an existing phone file. Ctrl-Y will tell you what record it's currently processing.

FONEBULD will check for bad numbers, i.e., non-existent area code, and issue an error message like the following example:

```
EX:
"Invalid phone number (999 555 1212), will not add."
```

800, 888, and 900 phone number area codes will be set to time zone 6 unless you override this in columns 23-24 of the raw record, or change the ZONETABL using the MAKEZONE utility (see *5.13 MAKEZONE*).

Unless DUPLICATES OK has been set to YES in the shop file, duplicate numbers will be resolved with status 91. These numbers will be put in the phone file but never be called. To keep a duplicate number OUT of the phone file, see example file DUPES^SPX, or use FONEUTIL to delete the numbers after they have been built into the file.

FONEBULD will prompt for another study when you are done processing the first. If you type "newfile" at the prompt, it will prompt you again for commands to start building or adding to another phone file.

6.1.2 Adding Records to an Existing Phone File

FONEBULD also allows you to add more phone numbers to an existing phone file, or to rebuild a phone file once it has been converted to ASCII (but still keep histories, etc.). To add new numbers to an existing file, use the command GO, enter the study name, and then enter the file name of new numbers.

The ASCII option of FONEBULD allows you to rebuild phone files from records converted to ASCII by FONEUTIL or Mentor. This will allow you to retain the original phone histories of these records. It also means you can edit the records and read them to a phone file, or convert records from some other study to be continued in your new study.

To rebuild a phone file or add records converted from some other study, use the ASCII option, enter the name of the study, say GO, then enter the name of the converted ASCII file.

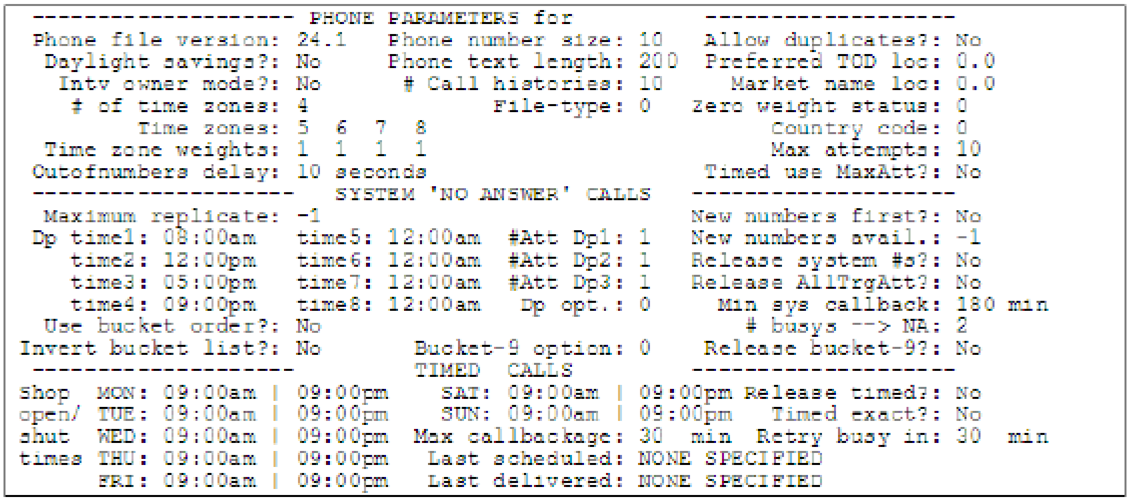
6.2 THE PHONE FILE

When the phone file is built, the parameters are loaded into the top of the file followed by the phone records. The records are given a record number and linked together in "stacks" with pointers that point from one record to the next record in its stack. The number of phone records allowed per study is 10 million. A phone file can now have up to 4.28 gigabytes for systems that support files greater than 2.1 gigabytes.

The records contain detailed information including what was initially loaded, information on the status of the number, and a call history for each call. The following is a description of the parameters in the file that control how Survent will retrieve phone numbers and a description of the fields recorded in the phone file.

6.2.1 The Phone File Parameters

When you run FONEBULD, the initial set of phone file parameters are read into the phone file. These control how Survent behaves when it goes to retrieve phone numbers for calling. They can be modified interactively in FONEBULD using the SHOPFILE screen, in SURVSUPR while a study is live, or in FONEUTIL when a study is not live. Each parameter also has a command that can be used to change it. The set of commands to create the parameters can be created using the HEADER command in FONEBULD or FONEUTIL, which may then be edited and read into any of these programs. Here is the standard parameters screen:



```
 --------------------- PHONE PARAMETERS for         --------------------
 Phone file version: 24.1   Phone number size: 10   Allow duplicates?: No
  Daylight savings?: No     Phone text length: 200  Preferred TOD loc: 0.0
   Intv owner mode?: No      # Call histories: 10     Market name loc: 0.0
    # of time zones: 4              File-type: 0   Zero weight status: 0
         Time zones: 5  6  7  8                          Country code: 0
  Time zone weights: 1  1  1  1                          Max attempts: 10
 Outofnumbers delay: 10 seconds                        Timed use MaxAtt?: No
 --------------------   SYSTEM 'NO ANSWER' CALLS   --------------------
  Maximum replicate: -1                            New numbers first?: No
 Dp time1: 08:00am   time5: 12:00am  #Att Dp1: 1   New numbers avail.: -1
    time2: 12:00pm   time6: 12:00am  #Att Dp2: 1   Release system #s?: No
    time3: 05:00pm   time7: 12:00am  #Att Dp3: 1   Release AllTrgAtt?: No
    time4: 09:00pm   time8: 12:00am   Dp opt.: 0     Min sys callback: 180 min
  Use bucket order?: No                                 # busys --> NA: 2
Invert bucket list?: No       Bucket-9 option: 0    Release bucket-9?: No
 --------------------         TIMED  CALLS         --------------------
Shop  MON: 09:00am | 09:00pm    SAT: 09:00am | 09:00pm Release timed?: No
open/ TUE: 09:00am | 09:00pm    SUN: 09:00am | 09:00pm   Timed exact?: No
shut  WED: 09:00am | 09:00pm Max callbackage: 30  min  Retry busy in: 30  min
times THU: 09:00am | 09:00pm   Last scheduled: NONE SPECIFIED
      FRI: 09:00am | 09:00pm   Last delivered: NONE SPECIFIED
```

To change a setting, use the arrow keys (monitor) or **Ctrl-U** (for up on a terminal; use **D** for down, **R** for right, and **L** for left) to position the cursor on the item, and enter the new value. When you are finished, press **ESC** to exit.

If you wish to change these after starting a study, you must build a new phone file.

What follows are explanations of the items included on the parameters screen. After each description the corresponding command is given with a sample value. An asterisk (*) before an item means that that option can be changed with a SURVSUPR MPF or FONEUTIL M command once the phone file is built.

GENERAL PARAMETERS

FILE VERSION The version of the software and phone file. The software version is for major changes in the structure of the phone file and is hard-coded in the software. The phone file version is for clients who need different dialing patterns. If set to 3, if a scheduled callback is to occur within Minage time and the Minage spans a day part change, and the callback results in a No Answer or Busy, the attempt will not be counted towards the number of calls morn/aft/eve or Max calls. If a Busy callback results in another busy and the prior attempt occurred before a daypart change, treat it like a No Answer in the original day part. (PHONESYSVERS=3)

DAYLIGHT SAVINGS Controls whether the system should adjust calling times for Daylight Savings Time. When set to Y(YES), DST is in effect. This will cause the program to make the appropriate time adjustment for those areas that do not observe daylight savings time. For example, Arizona, which is in the Mountain time zone and does not observe daylight savings, will have all its numbers act as though they were in the Pacific time zone. ***NOTE:*** Daylight Savings Time works only for North American phone numbers. See FONEUTIL options DS+ and DS- to change this once the phone file is built. (DAYLIGHT=Y/N)

***INTV OWNER MODE** Specifies whether you are using interviewer ownership mode in this study, where numbers can be specifically assigned to particular interviewers. The default is NO. (See OWNERSHIP MODE discussion below) (OWNERMODE=Y/N)

# OF TIME ZONES The number of time zones into which phone numbers may fall. The maximum is 8. They need not be consecutive (NUMTIMEZONES=4)

TIME ZONES The time zones used. A time zone is specified as a number, where 24 is Greenwich Mean Time. The zone to the west is 1 and each subsequent zone to the west is incremented by 1, up to 24. Time zones 5, 6, 7, and 8 are Eastern, Central, Mountain, and Pacific times in North America. Alaska is in time zone 9 and Hawaii is in time zone 10 during non-daylight savings time and is in 11 during daylight savings time. Time zones 22, 23, and 24 are for Europe. You must set at least one time zone before exiting. The time zones used do not have to be consecutive (TIMEZONES=5-8,10).

***TIME ZONE WEIGHTS** Used to change the proportion of non-timed dialings made to each time zone. Setting WEIGHTS= 2 1 1 1 would cause the first time zone to be called twice as often as would otherwise be expected (see *6.4.2 How Survent Chooses a Phone Number*). Setting a given time zone's weight to zero will exclude that time zone from attempts, except for specific-timed callbacks, Weights can be integers from 0-9. (TIMEZONEWGHTS=11112222)

***OUT OF NUMBERS DELAY** Number of seconds to pause between attempts to get a number by an interview when the system has run out of numbers. The accepted range of values is 0-99 seconds. The default is 10 seconds. (OUTOFNUMDELAY=20)

PHONE NUMBER SIZE Maximum number of digits in the phone number. The accepted range of values is 5 to 19. If the country code is “0” and the phone number size=10, all numbers must have 10 digits (PHONENUMSIZE= 10)

PHONE TEXT LENGTH The amount of user text the phone file is to hold. This number may be up to 4900 in version 7.6l or greater, 900 for prior versions. (TEXTLEN=400)

# OF CALL HISTORIES The maximum number of call attempt results saved for a number. A history includes a call's start time, seconds on the call, the interviewer ID, the call number, and its assigned status. If more calls are made to a number than allocated here, call result information will be lost. The program will save the first attempt and the last N-1 attempts where N is this setting. The maximum value here is 99. (NUMHIST=10)

FILE-TYPE Can be set to any number 1-99 and is used for just informational purposes. (FILETYPE=3)

ALLOW_DUPLICATES Controls whether to check the phone number or the other 2 indices in the phone file for duplicates. If set to Yes (YYY is “yes” for each index), it will allow duplicates in all the indices the phone file. If set to No (NNN), duplicates will not be allowed in any of the indices. The three characters are “Ys and “Ns” where “Y” means duplicates are allowed for that index, and “N” means they are not. ALLOW_DUPS=NYY means duplicates are disallowed for the phone number index, but allowed for the other two indices. Numbers that are duplicates are resolved with status 91.

NOTE: You may get duplication in your file later if you allow phone numbers or the other indices to be changed during interviewing. (ALLOWDUP=YYY/NNN)

The allowable settings are:

NNN	duplicates not OK for any index.
NYN	dupes (duplicates) not OK for phone numbers OK for index 1.
NYY	dupes not OK for phone numbers but OK for index 1 and 2.
YNN	dupes OK for phone numbers but no index files.
YNY	dupes OK for phone numbers and for index 2, but not index 1.
YYN	dupes OK for phone numbers and index 1, but not index 2.
YYY	duplicates OK for all indices.

As always, numbers rejected due to duplication get status 91, and if you have an allow_duplicates file, those numbers are not checked for duplication.

PREFERRED TOD LOCATION This tells FONEBULD which bucket to use when scheduling system calls (1-9); you may specify a different order for groups of records or all records, allowing independent call scheduling for each phone record. You can fill the field specified with the values to control the order before the file is built, or “on the fly” during interviewing by using “!PHONE,P” statements to fill the proper field before the system schedules the next call. This does not affect the first call to the number, because this is controlled by the initial call state setting in the raw phone record (columns 31-45). *See Section 6.6.3* for more information on how to use preferred time of day sampling. (PREFTODLOC=51.5)

MARKET_NAME_LOC This indicates where the market name is in the phone text area. Specifying markets allows you to control the release of sample by the number of numbers in the market and by using time zone weights. Small markets will be released slower than large markets. The market name consists of a position and width, the position must be in the phone text area and the width must be 1-8. (MARKETLOC=51.4)

ZERO WEIGHT STATUS Says what status to give to timed numbers that attempt to come up when the market weight for their market is set to “0”. Because some statuses are used by CfMC and some statuses cause inappropriate actions, this is limited to the following statuses:

Release number status: 0

Resolved statuses: 10-70

Nonspecific callback statuses: 107-156 or 191-199

Nonspecific callback statuses/No history: 201-209 or 213-214

NOTE: This is the status a timed number will get if it comes up while it's market weight is "0". If it is set to 0, no status is given and the number comes up to be called. (MARKETWEIGHT_ZERO_STATUS=70)

INTERNATIONAL_DIALING_OPTION If you set the country code to a number greater than 0 in FONEBULD, Survent will not check for valid North American numbers when doing Get_specific mode and PHONE,C to change the phone number. You can change the country code for specific phone records using the INTERNATIONAL_DIALING_OPTION field in the phone record layout. Phone numbers are displayed without extra formatting characters, i.e., (415) 777-0470 is displayed using INTERNATIONAL_DIALING_OPTION=0 but it is displayed as 4157770470 otherwise. (COUNTRY=0)

***MAXIMUM ATTEMPTS** Specifies the number of calls to make to a phone number. This setting can be changed on the fly in SURVSUPR and in FONEUTIL. This number should be >= the sum of attempt1, attempt2, and attempt3. By default, timed calls continue to be called beyond “max attempts,” unless you use timed status 184, which reschedules at the time specified unless maximum attempts are reached, in which case it resolves the number with status 94 such as system calls.

Calls that get a non-timed status and are a call >= the max calls setting will get a status of 94 in the “final status” field (but their assigned status in the “last call status” field) and be resolved

unless USEBUCKET9=2, in which case, calls that would get status 94 are put into bucket 9 to be released later. The maximum value is 999. (MAXATT=15)

TIMED USE MAX ATT If set to “Yes,” this tells the program to resolve timed numbers with a status of 94 when they are called on the “last attempt” time. The default is that timed numbers get rescheduled instead of being resolved (TIMED_CALLS_USE_MAXIMUM_ATTEMPTS_ATTEMPTS=NO)

SYSTEM CALL PARAMETERS

These parameters control the release of new numbers and rescheduling of “no answer” callbacks.

***MAXIMUM REPLICATE** If ‘replicates’ have been assigned in your phone records, then only those new numbers with a replicate number less than or equal to MAXIMUM REPLICATE will be released to the system. For example, if you set this to 3, then any new phone number with a replicate number of 4 or more will not be released. Once a number has been dialed, it is no longer under replicate control. If set to -1, then replicate control is ignored. This cannot be changed to/from –1 once the phone file is built (MAXREPLICATE=1)

***DAY PART TIME 1** The beginning of the day in respondent time for system scheduled calls. This also defines the beginning of the weekend day if day part times 5 and 7 are not used. No respondent will ever receive a system scheduled call before this time. (DAYPART1=11:45am)

***DAY PART TIME 2** The start of the second time period and the end of the first on weekdays for system scheduled calls. The time is respondent time. DAY PART TIME 2 must be later in the day than DAY PART TIME 1. (DAYPART2=1:25pm)

***DAY PART TIME 3** The start of the third time period and end of the second on weekdays for system scheduled calls. The time is respondent time. DAY PART TIME 3 must be later in the day than DAY PART TIME 2. (DAYPART3=3:30pm)

***DAY PART TIME 4** The end of the day during the week. This also applies to weekends if day part times 6 and 8 are not used. The time is respondent time. DAY PART TIME 4 must be later in the day than DAY PART TIME 3. No respondent will ever receive a system scheduled call later than this time. (DAYPART4=6:25pm)

***DAY PART TIME 5** The start of the day for Saturday. This will also be the start of the day for Sunday if day part time 7 is not used. Set both day part times 5 and 6 to 12:00 am to not use them. (DAYPART5=10:00am)

***DAY PART TIME 6** The end of the day for Saturday. This will be the end of the day for Sunday if day part time 8 is not used. (DAYPART6=6:00pm)

***DAY PART TIME 7** The start of the day for Sunday. Set both day part times 7 and 8 to 12:00 am to not use them. (DAYPART7=11:00am)

***DAY PART TIME 8** The end of the day for Sunday. (DAYPART8=5:00pm)

NOTE:

- Any of the day parts 5-8 may be missing but if only 5 and 6 are set, those times will be used for both Saturday and Sunday.
- If all of day parts 5-8 are missing, day parts 1-4 will work as they did previously.
- Using the Time Period Option (TPO) will affect how day parts are handled.

USE BUCKET ORDER Says whether or not to use a custom bucket priority order for calling system scheduled callbacks. If set to Yes, a prompt will appear, requesting the bucket order. The default is that system scheduled callbacks are called using bucket order 1-6.

BUCKET_ORDER=#,#,#,#,#,#,#,#,#,# Determines the priority order of buckets to look for system callbacks to be called.

The default is 1-7 then 0, with buckets 8 and 9 having special controls.

This means calls that have had the most system attempts have the highest priority. If you say "BUCKET_ORDER=0,7,6,5,4,3,2,1", new numbers will have the highest priority, then numbers

available in all time periods, then numbers in some periods, and finally numbers only available in this period. (BUCKETORDER=1,3,5,2,4,6,7)

***INVERT BUCKET LIST** If set to YES, this will reverse the bucket order search so that instead of looking for numbers in the call specific buckets first (1-3), then the call in some time period buckets (4-6), and then the call in any time period bucket (7), the program will look first in bucket 7, then in 6-4, and then in 3-1. (INVERTBUCKETLIST=Y)

***# CALLS PER DAY PART** Three numbers specify the number of attempts to make in each time period. TARGET= 1 1 2 says make one attempt in day part 1, one in day part 2, and two in day part 3/weekend. By default, numbers called on the weekend are called any time with no day parts. The order in which the attempts are made cannot be specified unless using preferred time of day sampling, which overrides targets, except that the first attempt can be in a specific time period if you use BUCKET=# in the phone record columns 31-45 to initially put the number into a specific bucket. If a call's length causes it to go across dayparts, it counts as an attempt in the earlier day part. (TARGET1=1, TARGET2=3, TARGET3=2)

***TIME PERIOD OPTION (T.P.O.)** The time period option controls how the system handles the scheduling of No Answer callbacks. The default is that there are three time periods called:

- morning (or day part 1)
- afternoon (or day part 2)
- evening (or day part 3)/weekend

If you assign one call to be made in each period, then if you call in the morning the first time, the number will be assigned to the 'call afternoon or evening/weekend' category. The next call will only occur in those periods. If the next call occurs on a weekday afternoon, it reschedules for an evening/weekend call. Note that timed calls get counted as having occurred in their time period also.

Here are the Time Period Option (TPO) settings:

These settings correspond to the DP option which is located under the number of attempts (ATT # DP1, 2, 3 etc.)

TPO=0 (or 1) is the default setting described above, where morning/day part 1 is between day part time 1 and day part time 2, afternoon/day part 2 is between day part time 2 and day part time 3, evening Monday-Friday is between day part time 3 and day part 4, and weekends are Saturday-Sunday between day part time 1 and day part time 4 anytime. See section on day part times for setting start and stop times for the weekend.

time1 <= day part 1 < time2 (Monday-Friday morning)

time2 <= day part 2 < time3 (Monday-Friday afternoon)

time3 <= day part 3 < time4 (Monday-Friday evening, or

Saturday-Sunday any time (start time-end time))

TPO=2 causes the program to treat all seven days of the week the same; whatever you set your times to, those times are used the same for each day of the week.

time1 <= day part 1 < time2 (Monday-Sunday morning)

time2 <= day part 2 < time3 (Monday-Sunday afternoon)

time3 <= day part 3 < time4 (Monday-Sunday evening)

TPO=3 causes the program to treat each weekday as if it only had two day parts, any calls scheduled for day part 3 will be called on the weekend only. See section on day part times for setting start and stop times for the weekend.

time1 <= day part 1 < time2 (Monday-Friday morning)

time2 <= day part 2 < time3 (Monday-Friday afternoon)

time3 <= day part 3 < time4 (Saturday-Sunday any time)

Saturday-Sunday any time (start time-end time))

TPO=4 treats each weekday as having three time periods, but Saturday and Sunday are ignored.

This is for weekday-only calling.

time1 <= day part 1 < time2 (Monday-Friday morning)

time2 <= day part 2 < time3 (Monday-Friday afternoon)

time3 <= day part 3 < time4 (Monday-Friday evening)

TPO=5 treats the weekend days as having three time periods, but Monday to Friday are ignored.

This is for weekend-only calling. (TPO=5)

time1 <= day part 1 < time2 (Saturday-Sunday morning)

time2 <= day part 2 < time3 (Saturday-Sunday afternoon)

time3 <= day part 3 < time4 (Saturday-Sunday evening)

***BUCKET 9 OPTION** "Bucket 9" is an extra bucket in each time zone where numbers can be saved to be released using RELEASE_BUCKET_9. Numbers can be saved there either by giving them phone status 189 or 214, pre-assigning them in the phone record using the BUCKET=9 command in columns 31-45, or using this USE_BUCKET_9 feature as follows:

(USEBUCKET9=1)

1= numbers getting status 106 (predictive dialer nuisance calls)

2= maximum number of attempts have been made to a number

3= Both 1 and 2

4= used to ignore minimum system callback time. Users that have markets who want their timed numbers moved to bucket 9 when the market weight is zero, and called as soon as an interviewer is available to call that market.

Note that the 3 bucket 9 options (1,2 and 4) can be combined so they get used in tandem. For example, Use_bucket_9=6 would cause numbers hitting the max attempts to be put in bucket 9, and those numbers would ignore the minsyscallbacktime. This is the item called "Bucket 9 option:" on the shopfile screen.

You can also retrieve these numbers using "get-specific" mode (*see the !PHONE,5*).

***NEW NUMBERS FIRST** If set to Y(YES), will cause new numbers to come up before system scheduled callbacks. Otherwise, callbacks have a higher priority than new numbers. You may also set this to a percentage value from 1-99 to have new numbers get an xx% chance of coming up first, but you may only do this using the keyword, not on the SHOPFILE screen, eg.

FRESHFIRST=80. This is for cases where you want to "blend in" new numbers with the scheduled callbacks. (FRESHFIRST=Y/N)

***NEW NUMBERS AVAILABLE** Controls the availability of new (never called) numbers. The allowable range is -1 to 1000000. The default is set to –1; this setting is ignored, allowing an unlimited number of new phone numbers to be used by the system.

If greater than 0, each new number that comes up decrements NEW NUMBERS AVAILABLE by 1; when it reaches 0, new numbers will no longer be called. This allows you to call the first 100 new numbers, then stop calling new numbers until

these are called again (don't retrieve any more), then you can set it to 100 again, etc. If set to 0, new numbers will not be used. This allows you to force calling of callbacks only. (NUMFROMFRESH=2000)

***RELEASE SYSTEM #s** This allows calls to all system-scheduled callbacks even though they may only have attempts left in other day parts. For example, suppose you are trying to complete a study at night, but have no new numbers left, and have exhausted all your evening attempts on existing numbers. By setting this item to Y(YES), you would make numbers that only had morning or afternoon attempts left available now. (RELEASESTACKS=Y/N)

***RELEASE ALL TARGETED ATTEMPTS** When a number has had all day part attempts (as specified in DAY PART TARGETS), but has not reached the maximum number of calls (as defined in MAXIMUM CALL ATTEMPTS), it is placed in the ATA (all targeted attempts) bucket (bucket 7).

It remains there until RELEASE ALL TARGETED ATTEMPTS is set to Y(YES). This can be used to make three attempts on a number now, but save it so you can make a fourth attempt later on, if you are running low on available numbers. (ATA=Y/N)

***MIN SYSTEM CALLBACK TIME** The minimum time in minutes before a number scheduled as a 'no answer' can come up to be called again. Numbers scheduled for callback at a specific time by the interviewer are not subject to this restriction. The accepted range of values is 0-9999 minutes. (MINAGE=100)

NOTE: Also refer to parmfile option MINAGE_LIMITS in *Chapter 4*, which controls what users can specify for minimum age limits of "no answer" call backs.

# BUSIES NA This setting allows you to control how busies are treated. This controls two things: first, how many busies will be allowed before treating the number as a No Answer (call status 103); second, whether to count the intermediate busy calls toward the total number of calls made.

The value specified can be any number from -9 to +9. 0 means that each busy is given a busy status and treated as one call. A negative number says how many busies in a row to allow before getting a status of No Answer, with each call counting as one attempt. A positive number says how many busies in a row to allow before getting a status of No Answer, but the intermediate busy calls will not count towards the maximum number of attempts.

The default is 2 in the standard phone system; that is, two busies in a row are treated as one No Answer, with this only counting as one call for the purpose of counting attempts. (NUMBUSYNOANSWER=0)

For example, If NUMBUSYNA=4 and you have two busies and one no answer, you get NUMATTEMPTS+1, NUMCALLS=3, histories 102/102/101. The first two busies are not counted as attempts. If NUMBUSYNA=4 and you make four busy calls, you get NUMATTEMPTS=1, NUMCALLS=4 and status 102/102/102/103.

***RELEASE BUCKET 9** A number put into bucket 9 stays there until this setting is changed to YES. If YES, bucket 9 will be the first bucket the program gets numbers from, unless INVERTBUCKETLIST is also set to YES. (Releasebucket9=Y)

TIMED CALL PARAMETERS

These parameters control the scheduling and release of specific-timed callbacks and some additional parameters to control release of saved numbers and bucket priority list inversion.

***SHOP OPEN <day>** Indicates when the shop is opened, machine time, for a particular day of the week. Timed numbers cannot be scheduled for callback before this time, they can only be scheduled between the open and shut time for each day of the week. These times are independent of the DAY PART TIMEs and in no way affect the current availability of numbers. (OPENmon/tue/wed/thu/fri/sat/sun= 0800)

***SHOP SHUT <day>** Indicates when the shop is closed, machine time, for a particular day of the week. Timed numbers will not be scheduled for callback after this time, they can only be scheduled between the open and shut time for each day of the week. These times are independent of the DAY PART TIMES and in no way affect the current availability of numbers. (SHUTmon/tue/wed/thu/fri/sat/sun= 0800)

OPEN and SHUT times are both machine time, that is, the time in the time zone where the computer resides. When OPEN and SHUT times are both "12:00 am," the shop is always open that day; when both are "12:00 pm," the shop is always closed (after midnight the prior day). If SHUT time is less than OPEN time, then it is assumed to be that time the following day. If the next day is closed, you cannot set the SHUT time to be after 12:00 am.

***MAX CALLBACK AGE** The maximum time in minutes beyond the scheduled callback time that a call can still be made. When exceeded, the call is rescheduled for the same time the following day and is moved immediately to stack 313 (timed later calls). If that time is not within the open/shut and day part times, it is rescheduled as a timed call in the next available day part time. Since scheduled callbacks are of high priority, this is unlikely to happen unless your shop was not open during the original scheduled time or many numbers are all scheduled for the same time. NOTE: If MAX TIMED CALLBACK AGE is set to a negative number, and a number is not called by that time, it is resolved with status 95. (CALLBACK=10)

***LAST SCHEDULED** The last date/time timed callbacks can be scheduled for. You will not be able to schedule a time beyond this date/time (LASTSCHEDULED=4/30/95)

***LAST DELIVERED** The date/time after which phone numbers for this study cease to be released. (SHUTOFFTIME=4/30/97)

***RELEASE TIMED CALLS** This releases specific time callbacks to be called now even though they are scheduled for later than now. If a number gets a busy status when this is set, it is given status 182 and treated as a No Answer call. If it is given a timed status, it is saved with status 105, so timed numbers won't come up again right after being called. Calls are released in callback time order. (RELTIMED=Y/N)

***TIMED EXACT** Tells the program whether to try and make timed calls at EXACTLY the minute they are scheduled for, or whether to do the default of calling within 5 minutes of the scheduled time (EXACT_TIMED_CALLS=NO).

***RETRY BUSIES IN** The number of minutes from the last call that a busy number will be rescheduled for re-dialing. The maximum is 32000. Consecutive busies act as a No Answer depending on the value of NUM_BUSY_TO_NA. A (0) setting will cause busies to always act like No Answers. If a number is given a busy status, but the 'busy reschedule' time would occur after the "CLOSED" time, the number is given status 182 and scheduled to be called like a No Answer (i.e., in the next bucket time it would be available to be called in).

If the time specified is negative, for example -15, callbacks will be attempted every 15 minutes until you reach the respondent or give a different status. This is so you can get to those very busy respondents who are on the phone constantly. In this case, each call is treated as an attempt, but the number is not resolved when the maximum number of attempts is reached until it gets some other non-timed status. (BUSYRESCHED=30)

OTHER PARAMETERS NOT SPECIFIED ON SHOPFILE SCREEN

HISTORY_SLOT_LENGTH this tells the program what the length of the call history information area for each call is. This is version dependent, the minimum is 27 and the maximum is 100; currently this can only be set to 27, 34, or 100.

NOTE: SURVSUPR option MPF and FONEUTIL option M can change starred settings in the phone file once it is built.

PARMFILE PARAMETERS USED BY THE PHONE SYSTEM

The CfMC server reads a file called PARMFILE in the CFMC CONTROL directory when starting up. This will set some phone parameters that may be overridden for a particular study or some that may ONLY be set for the whole system. Here is a list of the PARMFILE parameters used in the standard phone system:

DO_NOT_CONTACT_FILE: tells the name of the file for FONEBULD to check to disallow certain phone numbers from being added into any phone file. The file must be sorted by phone number and may have up to 70 columns of associated comments. Phone numbers that are marked as “forbidden” are saved in stack 331 of the phone file. (DO_NOT_CONTACT_FILE: BADNUMS). *See Section 6.6.2, “The Do Not Contact File,”* for more information.

END_OF_DAY: this determines what time-of-day the program will use when it decides to move timed numbers from the "tomorrow" stack to the "current" days timed stacks. For instance, if you have "END_OF_DAY: 3:00" in the parmfile, then numbers whose scheduled callback time is 2:00 am will be move to tomorrows timed stacks if a call was scheduled for 2:00am the following morning.

Similarly, if the CfMC server is running at 3:00am on that study, the phone file will be integrated following the usual rules.

This command also causes the server's log file (l<a-l><date>) to be closed and a new one started so all log records for a particular date are named with that date.

Also, it closes the "lp<date>" log of ASCII phone records if "server_write_ascii_phone_records: Yes" is set for the same reason.

HARD_BUCKET_SCHEDULING: this disallows interviewers from scheduling timed calls outside of starting and ending daypart times for a study (daypart 1 and daypart 4 times generally). If this is not set, interviewers will still get a warning if they try to schedule calls outside of the bucket times. (HARD_BUCKET_SCHEDULING: YES)

KEEP_BEFORE_PHONE_9: YES/NO tells Survent what to do relative to questions asked prior to the !PHONE,9 statement when resuming SUSPENDED interviews. The default is that the original (old) responses are replayed when the resume occurs (when KEEPBEFOREPHONE9:NO is used). If you use the "YES" option, it will save the responses given in the current interview instead of the original. ***NOTE:*** Be careful when using this because it could affect questionnaire logic if you say "YES" and questions in the main part of the questionnaire are based upon actions done by questions prior to the !PHONE,9.

HARD_BUCKET_TIME: this tells the server whether to treat the starting and ending daypart times for a study (daypart 1 and daypart 4 times generally) as the ONLY time that calling may be done. If timed calls or special interviewer numbers come up outside of daypart times, and this parameter is set to "YES", the numbers will be put back in the stack they came from. (HARD_BUCKET_TIME: YES)

MARKETWEIGHT_ZERO_STATUS: this is the same as the **ZERO WEIGHT STATUS** parameter above; it set the default status to return for this server if ZERO WEIGHT STATUS is not set in the study. Because some statuses are used by CfMC and some statuses cause inappropriate actions, this is limited to the following statuses:

Release number status: 0

Resolved statuses: 10-70

Non-specific callback statuses: 107-156 or 191-199

Non-specific callback statuses/No history: 201-209 or 213-214

NOTE: This is the status a timed number will get if it comes up while it's market weight is "0".If it is set to 0, no status is given and the number comes up to be called. (MARKETWEIGHT_ZERO_STATUS=70)

FONEBULD with load MARKETWEIGHT_ZERO_STATUS value from the parmfile if none is specified. If MARKETWEIGHT_ZERO_STATUS is specified in the parmfile, the value will be copied into the header of the fonefile that is being made. The MARKETWEIGHT_ZERO_STATUS command in fonebuld overrides the value in the parmfile.

SERVER_WRITE_ASCII_PHONE_RECORDS: this command saves an ascii copy of the phone record each time you write a status to the phone file. The purpose is so that you can get realtime phone reports or transfer the data to other systems without having to convert records from your phone files.

The server writes to a file LP<ddhhmm> (DOS) or lp<ddhhmm.yyyy_mm> (UNIX) in whatever directory the server is running from (###### is the day, hour, and minute the file was created, yyyy is the year and mm the month).

If you type "server:log" at a supervisor, the server closes the current lp###### file and starts a new one, so you would want to do this just before running your application. Also, the "END_OF_DAY: HH:MM" parameter will cause the program to start a new “lp” file so that each "day"s files has all the records for that day.

SPECIAL_ONLY_SPECIAL: ”YES” says that interviewers logged on with “special” interviewer types will ONLY get numbers assigned to those type(s); the default is that “NO” which means that if there are no special interviewer numbers available those interviewers get numbers from the regular stacks. Note that this may cause interviewers to have to wait for more numbers to be transferred to the “special” stacks.

TIME_ZONE: specifies the “local” time zone for your server to determine when to make calls to numbers. This is a required parameter for Survent. (TIMEZONE: 05)

UPDATE_FONEHEAD: tells the program to update phone header information for a study as soon as the user modifies the parameters in the supervisor. The default is that the program waits until the next phone update. (UPDATEFONEFONE: YES)

PARMFILE PARAMETERS USED IN LARGE PHONE FILE SYSTEMS:

ASCII_PHONE_REC_TYPE: this tells FONEBULD and FONEUTIL how to read and write converted ASCII phone records in LARGE phone file versions. It may be set to "SQUISH" or "RAW". If set to "SQUISH", instead of writing out all the columns in their standard positions (which amounts to 10,000 columns per record), FONEUTIL puts out the phone text in the smallest multiple of 200 that will fit all the text, and it puts a letter from A-Z to specifies how many multiple of 200 the actual phone text length is. For instance, if you have a phone text length of 1200, the program will put an "F" (6th letter * 200) in column 20 of the ASCII record and only write the 1200 columns of text to the file (instead of the max of 5000). The call histories are similarly shortened to your "history length" instead of 100 columns. This file can then be properly read in FONEBULD when you use the ASCII option. If set to "raw," the program writes the data in its standard positions. (ASCII_PHONE_REC_TYPE: RAW)

HISTORY_SLOT_LENGTH: this specifies a REQUIRED call history length for this server. If any studies have a history length specified differently that this value, FONEBULD will not build the phone file for the study. This is used in LARGE phone file systems where call histories can have a variable length (34-100). (HISTORY_SLOT_LENGTH: 100)

FONE_HISTORY_START=### is used by clients who use nonstandard formats for the phone call history area. This allows them to specify valid locations in their questionnaire specs when referencing phone history variables.

PHONE_REC_LOC_STYLE: tells the phone system whether to interpret user references to CfMC phone variables as "actual" or "standard" location references in "LARGE" phone file systems. By "actual" we mean the position of the field in the record relative to its phone text length; if the phone text length is 1900, the "actual" position of the CfMC variables would start in position 2000. The "standard" position starts in column 5000, which is where the variables are if you use the biggest phone text length possible (4900). This affects !PHONE,G and !PHONE,P references in the questionnaire and the SELECT statement in the SURVSUPR and FONEUTIL programs. The default is “standard”. (PHONE_REC_LOC_STYLE: STANDARD)

PHONE_TEXT_LENGTH: like the HISTORY_SLOT_LENGTH parameter, this forces all studies used under this server to have the same phone text length for servers supporting LARGE phone file systems. The allowable maximum phone text length is from 200 to 4900. (PHONE_TEXT_LENGTH: 4900)

6.2.2 The Phone Record Format

In addition to the phone file parameters, to interact with the phone file you must know the phone record format. This is where the information about each phone record is updated and stored as Survent makes calls to the numbers. The places where you use this information most often are 1) Writing the questionnaire using the PHONE,G command to record sample information in the data file or the PHONE,P command to update the phone text area, 2) In SELECT statements in SURVSUPR and FONEUTIL when hiding or revealing sets of number of getting lists of numbers with certain criteria, 3) Converting files to ASCII to be modified and read back into a phone file using the ASCII command in FONEBULD, and 4) Running reports on the sample information at the end of the study.

The file is treated as fixed-column format. In many places you can reference the field by its variable name or data location, but sometimes only the location may be referenced.

The "Type" determines how the data is stored; type S for "string variable" means the data is left-justified in the field and treated as a string, **type N** for "numeric variable" means it is right-justified in the field with leading spaces and can be treated as a numeric field using CfMC software.

Here is the phone record format:

Location	DESCRIPTION	VARIABLE NAME	TYPE
1	Phone area code	AREA_CODE	
1.19	Phone number	PHONE_NUMBER	S
4	Phone prefix	PREFIX	
7	Phone suffix	SUFFIX	
20.2	Reserved for system		
22.1	Special interviewer type (1-9)	SPECIAL_TYPE	N
23.2	Time zone (01-24)	TIME_ZONE	N
25.10	Case ID to get for data case mode	CASEID	S
31.14	Callback time (yyyymmddhhmm)	CALLBACK_TIME	S
47.3	Replicate number (001-999)	REPLICATE	N
50.1	Reserved for system		
51.4900	Phone text	PHONE_TEXT	S
4951.49	Reserved for system		
5000.5	Total time in seconds on number	TIME_ON_NUMBER	N
5005.5	Number of attempts to number so far	ATTEMPTS_MADE	N
5010.5	Stack came from last call	PREVIOUS_STACK	N
5015.5	Time zone bucket (0-9; -1 if not in grid)	BUCKET	N
5020.5	Current calling stack number is in	STACK_NUMBER	N

```
5025.5          Stack was in when last hidden                      HIDDEN_STACK     N
5038.3          Length of call history field                       HISTORY_NOTE_LEN N
5041.3          Length of phone text area                          PHONE_TEXT_LEN   N
5044.1          Error stack flag, 1=yes, 0=no                      IN_ERROR_STACK   N
5045.1          1 if approx. time status                           APPROX_TIMED     N
5046.2          Phone file version                                 PHONEFILEVERSION N
**5048.8        Name of file to resume if suspended                RESUME_FILE      S
5057.1          Suspend flag, 1=yes, 0=no                          SUSPENDED        N
5058.5          If using Phone Says Datarecord mode, 1=yes         PHONE_SAYS       N
5063.5          Number of times phone file written to              NUM_WRITES       N
5068.5          Number of history slots in phone file              NUM_HISTORY      N
5073.2          Number of attempts, no day part                    ATTEMPT_0        N
5075.2          Number of attempts, day part 1                     ATTEMPT_1        N
5077.2          Number of attempts, day part 2                     ATTEMPT_2        N
5079.2          Number of attempts, day part 3                     ATTEMPT_3        N
5081.10         Case ID of complete(left-justified)                CASEID           S
5091.19         Old phone number,if changed                        OLD_PHONE_NUM    S
5110.1          Daylight time, 1=yes, 0=no                         DAY_LIGHT_TIME   N
5111.3          Final status of resolved call 0=unresolved         FINAL_STATUS     N
5114.3          Total number of calls                              CALLS_MADE       N
5117.3          Last history slot used                             LAST_HISTORY     N
5120.1          If changed phone number, 1=yes, 0=no               PHONENUM_CH      N
5121.2          State code (AL-WY)                                 STATE_CODE       S
5123.3          # of times changed phone number                    NEW_PHONE        N
5126.3          # of times cleared histories(w/ZAP/ERASE)          HISTORY_CLEARED  N
5129.3          Market number (0-255)                              MARKET_NUMBER    N
5132.3          Whether number is on do not contact list           DNC_FLAG
5135-5143       Unused space
5144.7          Record number in phone file                        RECORD_NUMBER    N
5151.1          Interviewer ownership mode:1=yes,0=no              OWNER_MODE       N
5152.1          Interviewer ownership changed:1=yes,0=no           OWNER_CHANGED    N
5153.4          Interviewer ID who owns number                     OWNER            S
5157.4          Interviewer ID,previous number owner               OLD_OWNER        S
***5161.15      Stack name of stack currently in                   STACK_NAME       S
5176.3          Country code (0 or 1=U.S/Canada,or > 1)            COUNTRY_CODE     N
5179.8          Name of file the phone record was added from (
                first 8 characters)                                RAW_FILE_NAME
5187.12         Date/time phone record added to phone file (YYYYMMDDHHMM) TIME_ADDED S
5199.8          First 8 characters of market                       SHORT_MARKET_NAME S
5207.2          Market weight when converted                       MARKET_WEIGHT    N
5209.2          Time zone weight                                   TIMEZONE_WEIGHT  N
5211.1          Whether number is "in-the-air"                     UPINTHEAIR       N
5214.3          Previous market number                             PREV_MARKET      N
5217.3          Number of times market was changed                 MARKET_CHANGES   N
5220.2          User Specified Phone file type                     PHONE_FILE_TYPE  N
5222            Long name of study                                 STUDYNAME
5252            Time added to last stack                           TIME_ADDED_TO_STACK
5270            Name of group hidden                               NAMED_HID

5300            Makes name of market available                     MARKET_NAME      S
5320            Set of 2-digit codes of last 8 operations          LAST_OPERATIONS
5352.28         Label of question where suspend occured            SUSPEND_LABEL
```

** The name will be left-justified if active, or moved to the right 1 space if it is an interview previously resumed. *** See following, STACK NAME Field in the Phone File, for actual stack names. The following is the information recorded for each call history. Version 7.7 uses up to a 100-column call history.

```
Information from last call recorded:                        LAST_CALL        S
6000.3           number of last call                        CALL_NUM_LAST    N
6000.100++       Position for each call history             HISTORY00-HISTORY99
6003.3           status of last call                        STATUS_LAST      N
6006.4           interviewer ID                             INTV_LAST        S
6010.14          call time (yyyymmddhhmmss)                 CALL_TIME_LAST   N
6028.5           # of seconds on last call                  CALL_SECS_LAST   N
6033.1           “Daypart” (1=morn, 2=aft, 3=eve/wknd)      DAYPART_LAST     S
6034.66          call note on last call                     CALL_NOTE_LAST   S

Call history for each call made (1-99):                     FIRST_CALL       S
6100.3           number of first call (001)                 CALL_NUM_01      N
6103.3           status of first call                       STATUS_01        N
6106.4           interviewer                                ID INTV_01       S
6110.14          call time (yyyymmddhhmmss)                 CALL_TIME_01     S

6128.5           # of seconds on first call                 CALL_SECS_01     N
6133.1           “Daypart” (1=morn, 2=aft, 3=eve/wknd)      DAY_PART_01      S
6134.66          call note on first call                    CALL_NOTE_01     S

++ Locations incremented by 100: 6000.100, 6100.100
```

Locations and names are repeated for other calls: (locations incremented by 100)

```
6100.3,6200,...,15900 number of first call         CALL_NUM_01,CALL_NUM_02,…,CALL_NUM_99
6103.3,6203,...,15903 Status of first call         STATUS_01,STATUS_02,…,STATUS_99
6106.4,6206,...,15906 Interviewer                  ID INTV_01,INTV_01,…,INTV_99
6110.14,6210,...,15910 Call time(yyyymmddhhmmss)   CALL_TIME_01,CALL_TIME_02,…,CALL_TIME_99

6128.5,6228,...,15928 first call, # seconds        CALL_SECS_01,CALL_SECS_02,…,CALL_SECS_99
6133.1,6233,…. .15933  Daypart,first call (1-3)   DAYPART_01, DAYPART_02, DAYPART_03
6134.66,6234,...,15934 Call note on first call     CALL_NOTE_01,CALL_NOTE_02,…,CALL_NOTE_99
```

Note that for the slots, the last history slot comes first, and then the slots are in order from call 1 to call n. This is so you can find the last call easily.

Additional Information about Phone File Fields

TIME_ON_CALL (5000.5) Total time in seconds on this call record, which is the sum of all calls from the time the phone record is retrieved until it is put back in the phone file. For individual call times, see the call history for a particular call.

ATTEMPTS_MADE (5005.5) This is the total number of attempts that count toward MAXIMUM ATTEMPTS, that is, once this value equals the MAXIMUM ATTEMPTS value and the current status is not a timed callback, the phone record will be resolved with status 94. Also for ATTEMPTS_MADE, by default 2 busies only count as 1 attempt. This is different from the CALLS_MADE (5114.3) field, which would count the 2 busies as 2 calls. Also, ATTEMPTS_MADE gets reset when you use the “Z”ap or “E”rase options in FONEUTIL, while CALLS_MADE does not, so you always know how many calls were actually made to the number.

STACK_ NAME (5161.15) The STACK_NAME shows the name of the stack the number is currently in. It matches the **STACK_NUMBER (5020.5)** but is in a more readable format. This is often used in reports; for instance, you can do a frequency report with Mentor’s ~FREQ command on this field to get a list of how many records are in each stack, or display the name when looking at a particular phone number.

It’s most useful feature, though, is that you can change the stack name and then add numbers into another file with a different version or time zone scheme and have the number go to the correct stack, without changing any other fields.

To read in numbers from an old phone file and change their stack without losing any history information:

In FONEUTIL, enter D to delete (if you will be reading back into the same file), or C to convert records (to copy to a new file).

Modify the STACK_NAME variable in the ASCIIfied phone file to the name of the stack you want it to be placed in. The name(s) to use are as follows:

Description:	**Stack #:**	**Keyword:**
Timed today	1-288	T####

where #### is the time in 5 minute increments. "T1205" would be for 12:05pm.

```
Approximate timed today 289-312 TAPROX##
```

where ## is the hour of the day 01-24.

Timed later	313	MANANA
Complete/resolved	314	COMPLETED
Error	315	ERROR
Duplicates	316	DUPLICATE
Hidden numbers	317	HID_(Some other keyword)

e.g. HID_MANANA came from the MANANA stack.

```
     Special Intvwr 1-9       318-326      SPEC-#
```

where # is for special interview types 1-9

```
     Owned by interviewer     327          OWNEDRECS
     'In the air' recs        328          INTHEAIR
     'Dummy' records          329          DUMMYDO
     'Forbidden' records      330          FORBIDDEN
     Bad time zone recs       331          BADZONE
     Deleted records          349          FREE
     System numbers           350-???      TZ##B?
```

where ## is the time zone relative to Greenwich (01-24), and B is the bucket (0-9). These numbers will be put back into the correct time zone for your file, regardless ofwhether you have changed the time zone scheme.

TZ05B0 would be a number in time zone 5, bucket 0.

```
     Numbers in markets       350-9600     TZ##B?Mmmm
```

The STACK_NAME for numbers in markets is TZ##B?Mmmm where ## is the time zone,? is the bucket, and mmm is the market number (1-255). TZ08B3M023 would be time zone 08, bucket 3, market 23.

Once you have changed the names to what you want, use the ASCII option of FONEBULD to read the records back in. The numbers will be placed in the appropriate stack.

INTERNATIONAL_DIALING_OPTION (5176.3) This field controls 2 things: Whether to check the timezone table for a proper area code/time zone, and how to format the phone number in displays. If the country code is '0' or '1', the zone table will check for valid area codes/time zones, and place the number in the proper time zone if not specified in the time zone field. Also, the number will be displayed like (xxx) yyy-zzzz, that is, area code in parentheses, followed by the prefix and then the exchange. Other country codes have their numbers displayed without parentheses or dashes, and are REQUIRED to have a time zone specified in columns 23-24 of the phone record.

LAST_CALL (6000.100) LAST_CALL information is a copy of the most recent call's call history. The information is placed in the

"LAST_CALL" field as well as the field for it's particular call number (#1, #2, etc.). This is often used for reports, for it tell the current status of the record.

6.3 WRITING THE QUESTIONNAIRE FOR THE PHONE SYSTEM

When using the phone system, the questionnaire must include PHONE statements. It may also include functions or other statements related directly to the phone system.

- PHONE statements with a number subtype (indicated afterwards as PHONE,#) can be used to get a phone number from the phone file, display it and its previous call history or display the default call status screen. One or more of these must be chosen if using the phone system.
- PHONE statements with a letter subtype get information from the phone file, make it available to the interview and allow you to put information back into the phone file. These are optional.

NOTE: PHONE statements that display text do not clear the screen first. To do so, use the following statement just prior to the PHONE statement:

```
{ ''clear the screen
\T
!DISPLAY,2}
```

6.3.1 The PHONE Statement

The syntax for the PHONE statement is:

```
!PHONE,subtype,<parameter1,…,parameter n>
```

PHONE,# SUBTYPES

0 Gets and displays the phone number, its call history (up to 5–last four and first), and the default status screen (see *6.4.3 Status Codes Returned From Survent*). This type is seldom used because clients want to design their own status screen, use a code list and a PHONE,4 and/or PHONE,5 statement (see 6.3.3 *Sample Prepare Specifications* Using PHONE Statements).

1 This gets a phone number to be dialed by an automatic dialer. This is so you can check quotas or such before sending the number to the dialer. See also PHONE,D.

2 Gets and shows the phone history and any suspend text if a number has been retrieved with an earlier PHONE ,# statement. Here is a sample call history display:

```
Call # 1: FRI DEC 5 1999 16:57 - 16:59 ID: TEST STATUS: NO ANSWER
Call # 2: TUE DEC 9 1999 10:23 - 10:28 ID: MARY STATUS: TIMED CALL
```

The default is to show all histories. You may specify two numbers that control the histories shown:

```
!PHONE,2, # from end, # from start, SHOWNOTE
```

This will let you see *from start* histories from the first call and *from end* histories from the last call made. If the *SHOWNOTE parameter is used, it will show the call note in addition to the call history information. Note that *SHOWNOTE can only be used if you have 100-column call histories, the last 66 columns are the note.

```
EX:
!PHONE,2,3,1
```

This will show the first call and the last three calls. If only one number is specified, that many histories from the end will be shown.

The text describing the phone status on the phone history can be controlled using the file *FONESTAT in the CFMC CONTROL directory. Each label can be up to 16 characters. Add a line to the file for every status you will be using consistently, or modify the text for existing statuses. If you do not provide text, the program will list the status number instead.

```
EX:
1: COMPLETE
```

```
…
13: DON'T RESPOND (example added status)
…
101: NO ANSWER
102: BUSY
…
113: LIKES TO TALK (example added status)
…
```

3 Shows the suspend text if the call had been suspended on the previous call. Note: You can also display the suspend text in the text of a question by using the syntax "\[20001.80]" in version 7.7 (see *2.5.3 DISPLAYING PRIOR RESPONSES AND DATA)*.

4 Gets the number only (no display). This is the standard way to get a phone number; see 6.3.3 *Sample Prepare Specifications Using PHONE Statements* regarding setting up a status screen after retrieving a number. If you place a PHONE, 5 statement before the PHONE,4, the program will prompt for a specific number to call when it sees the PHONE,4. Enter a number or press <Return> to get the next number only. See the example below.

5 Prompts for and retrieves a phone number. It is often useful to be able to get a specific phone number from the phone file, rather than one that the system automatically presents. For example, you may have 800 inbound lines set up for respondents to call in on at their convenience. The interviewer would then want to get that phone record when a particular person calls.

To get a specific number, you must have an indexed phone file (see *6.1.1 Building The Phone File)* and a PHONE,5 statement in your questionnaire. When Survent encounters the PHONE,5 statement, a switch is set so that when a PHONE,0 or PHONE,4 statement is encountered, the interviewer is prompted for a number to get from the phone file. If the interviewer presses **Enter** at this point, instead of entering a number, the system will get the next available phone number as if in usual mode.

Here is the standard prompt:

```
EX:
Enter number to call -->
```

If the phone system cannot find the requested number, the interviewer is asked to enter another. If the number found is either hidden or being interviewed on, a message will print to that effect and the interviewer will have to try another number.

The same will occur if the number is resolved or owned by some other interviewer or interviewer type, unless other statements are in effect to override this (see PHONE,6-8 and PHONE,N).

The syntax for the PHONE,5 statement is:

```
!PHONE,5,column,width,label or location
OR
<col>.2 !PHONE,5,column,width,label or location
```

The first syntax is used when getting a number in conjunction with the PHONE,4 statement. The "column,width" specification tells the program what text to display from the phone file case there are duplicate numbers to choose from. The duplicates are numbered on the screen and displayed for the interviewer to choose from. The idea is to display text such as their full name or address so the interviewer will know which record to choose.

```
EX:
!PHONE,5,101,50
```

This will display 50 columns of phone text information starting in column 101. Up to 650 duplicate numbers will be offered to the interviewer, who will look at the phonetext information displayed and pick the appropriate one. By default, the first 70columns of the phone text are displayed.

When looking at a screen of duplicate numbers, numbers will be shown that are hidden or resolved (and so not available) but they will be marked as "(hid)" or "(res)" on the screen.

The <label or location> is used to get a phone number using your own question display instead of the standard display. This allows you to do the following:

- You can put whatever text you want on a prior question when asking for the phone number. ***NOTE:*** You can use the VAR,P question to get data in phone number format; that is, the interviewer can enter numbers in any of the formats '1- (xxx)

xxx-xxxx' or such, and the program will strip everything except the phone number itself from the response.

- You can put phone numbers in a disk-based-response list in the questionnaire and have interviewers pick which phone number they want to call. When they pick the number, you give the disk-based-response question as the label of the phone number, and the PHONE,5 will retrieve that number. This way, the interviewer will avoid typing mistakes.
- You can have two additional indices with the phone file to get numbers with. The indices may be from 5 to 20 characters wide. To get a number by an index value, use a question to get the name/address/etc. desired from the interviewer (using a response list or an open-end), then send that response as the entry to look for in the PHONE,5. If the index value is not found, you will be prompted for a valid phone number as always. Whatever you use as the index must exactly match what is entered–you cannot use wild cards or subsets.

If you wish to prompt for another index value, see below.

If the second syntax above is used, and you put a two-digit location on the PHONE,5 question label line, the program will search immediately for the number, rather than waiting for a subsequent PHONE question. Also, a value will be returned to the data that tells you the results of the search. This value can be used to do different operations (e.g. using conditional statements to display different error messages). The values returned are as follows:

Value	Description
BM	Bad market name
NI	No index file. This aborts the interview.
NO	No more numbers
OA	Number is owned by someone else.
SB	Number is not valid (ie: too few digits)
SC	Specific number completed

Value	Description
SD	Number at dialer
SH	Specific number is in hidden stack
SJ	Specific number was in the BLOW stack
SN	Specific number not found
SR	Specific number resolved (but not completed)
SS	Number for a special interviewer
SU	Number currently in use by another interviewer
SW	Specific number wrong record (bug somewhere)
SX	Number not wanted on duplicates screen (reset and ask again)
SZ	Specific number was in “Do Not Contact” stack
TA	No number returned, but wait a while and try again later
<blank>	Good phone record returned

Using the indexing example above, if a good number was not retrieved, you could RESET to the question asking for a name/address/etc. and try again.

6 Programmatically executes the OVERRIDE keyword for Ownership mode. There are no parameters to this command (see 6.6.5 *Ownership Mode*).

7 Programmatically executes the NEWOWNER keyword for Ownership mode. There are no parameters to this command (see 6.6.5 *Ownership Mode*).

8 Assigns ownership of this phone number to the current interviewer. This writes the interviewer ID into the owner field of the phone record. There are no parameters to this command (see 6.6.5 *Ownership Mode*).

9 This is used with the PHONE,4 or 5 if you allow suspending of interviews. When acallback is made to a number that was suspended, by default the following occurs: questions prior to the PHONE,9 are executed, then, if contact is made and theinterview continues, the answers given so far this interview are dropped, the RESUME block is executed, and the suspended questionnaire's answers are re-executed, including those questions prior to the PHONE,9. An interviewer cannot SUSPEND prior to the PHONE,9. The PHONE,9 is normally placed at the end of the contact screen or screener, as soon as the proper respondent is on the phone.

The server parameter file PARMFILE has an option, "STUFFBEFOREPHONE9: NEW", which changes this slightly. When the RESUME occurs, it does not re-execute the suspended interview's questions prior to the PHONE,9, but uses the answers from the current interview prior to the PHONE,9 instead. There are no additional parameters to this option.

NOTE: Only one phone number will be retrieved per questionnaire. That is, once the system has picked up a number, it will not pick up another until another interview is started.

PHONE,(LETTER) SUBTYPES

These must have a previous PHONE,# statement (one that actually gets a number) before these can be executed.

PHONE,A writes a new phone record to the phone file via the CfMC server with information you specify. Here is the syntax and example for the PHONE,A question type:

```
!PHONE,A,<label or location of phone info>


EX:
!PHONE,A,[25.900]
```

Like the !PHONE,5 statement, If you specify a two-digit location on the !PHONE,A statement you will get a return code saying what happened to the record as follows:

```
{Label: .2 !PHONE,A,<label or location of phone info>}
```

The return codes are:

Value	**Description**
OK	Good number added
AC	Bad area code, not added
BB	Specific bucket bad, not added
BR	Column 21 of phone record not B, R or blank
BS	Special type is bad
BT	Time zone not in this phone file
DP	Duplicate number (if disallow duplicates)
FU	Fonefile doesn’t exist
ID	Case ID is bad
MK	Bad market name
NA	Phone number starts with 911
NB	Column 50 of phone record not blank
SC	Bad stack number
ND	Phone number not all digits
NZ	Couldn’t load zone table
OW	Owner name is bad
RP	Replicate number bad
TZ	No time zone and area code not in zone table
TX	Phone text too long
WL	Phone number wrong length, not added

This sends the specified data to the server where it will be treated just as FONEBULD treats an incoming raw number. If it has a valid format, the number will be added to the phone file. If it isn’t added, there will be appropriate error messages in the SERVER log file. Survent itself gets no messages back. ***NOTE:*** You cannot use this feature with Standalone Survent. You may set all the parameters described regarding loading raw phone numbers including the time zone, the special interviewer type, an associated case ID, a particular time to call, or a particular bucket or stack.

PHONE,A provides users with the ability to use four major features:

1 Random Digit Dialing

Using this scheme, the phone file is built with ‘x’ numbers. Once one of those has made its maximum number of calls without a complete, resolve that number with some status 2- but add one to the phone number and have Survent generate a new number with the PHONE,A. Do this up to 10 times until it you get a complete. Note that the generated number may want to keep some of the phone text from the prior number but not the call histories.

2 Generate new leads, ask about additional products

This is used to add a phone record for a new contact or because the client is willing to do a second interview on a new product. If either is true, get the phone number and any other information, add it to your current information, and use the PHONE,A to save the number to be called later. Or, if you would like to do another interview immediately, save the info with the PHONE,A statement, then copy the phone number to the “local scratch” area, start a new interview, get the number back from the scratch area, and call up the new phone record using a PHONE,5 statement (See SPECIAL types K and L in 3.1.5 *System Information Statements* for how to save data to the local scratch area, see also example file ADDPH^QPX).

3 Add numbers to an 800-number (inbound) dialing database

When a new person that is not in your current sample file calls, you can ask them for their name, address and phone number, etc., build a record, and add it to your phone file.

4 Add sample while a study is live.

This is accomplished by reading your raw sample file with a questionnaire to read the file and generate new phone records. The questionnaire would look like this:

```
~PREPARE COMPILE –SPECS

[PHONA,CASE_LENGTH=2000,QFF_FILE_NAME=ADDNM^QFF,FONE_TEXT_LENGTH=900]
```

```
{
!GOTO,AROUND } '' NEED TO HAVE A PHONE,4 STATEMENT TO NOTIFY
{!PHONE, 4 } '' THE PROGRAM THAT WE WILL BE USING PHONE FILES,
{AROUND: !GOTO } '' BUT DON'T ACTUALLY WANT TO GET A NUMBER SO SKIP
{RETURNCD: 5
!VAR,A,1 }
{FILENAME: 10.12 '' THIS IS THE NAME OF THE FILE OF NEW RECORDS
ADDNM^RAW " IT SHOULD BE PUT WHERE THE SERVER RUNS
!SPC,9 }
{'' THIS READS THE ASCII FILE AND GETS THE DATA FOR SURVENT
!ZSPC,11,6,[RETURNCD],[FILENAME],[81.950]}
{'' THIS PUTS THE DATA INTO THE PHONE FILE AS A NEW RECORD
!PHONE,A,[81.950]}
{!IF [RETURNCD^B] '' IF RETURN CODE IS BLANK WE GOT A NUMBER
!SPC,H } '' AND ARE DONE, SO ABORT AND GET ANOTHER
{Errtype: [RETURNCD] !IF RETURNCD$<>"N"
If error, display the problem
!FLD,A
A Doesn't have 2 arguments
B Bad subtype
O File open failure
N Nothing to read (No file or off end)
}
{
The return code is \:RETURNCD: due to \:Errtype
!DISPLAY,2 }
{!IF RETURNCD$="N"
You are either done adding phone records with \:FILENAME:
Or you gave the file \:FILENAME: did not exist or was not
In the right place. Congratulations or try again.
!DISPLAY}
~END
```

Then, whenever you have sample to add, copy your spec file, change the study name of this questionnaire to match the one you are updating (make sure the other header parameters match), and start a dummy interviewer on the study with questionnaire ADDNM^QFF. Once interviewing commences, the sample will be loaded as the SERVER finds time to load it (See example file PHONA^QPX to run and test this).

PHONE,B statement records a suspended file name to be used. Now, you can add a !PHONE,B statement to your !SUSPEND block to get the name of the suspend file written to the data file. The syntax is:

```
!PHONE,B,<label or location>
```

So, you insert {suspname: !phone,b} to store the eight-character name in the next available location as "suspname". or you can set up a question to write to and say {!phone,b,suspname} to write the name to an existing variable called "suspname".

This is useful for marking the record as suspended and matching the data record to the corresponding suspend file name in the sample file.

PHONE,C Replaces the phone number in the phone file with the one found in the question or data location referenced. If it is replacing the original phone number, the original is saved in the "OLD_PHONE_NUM" location in the phone file (see 6.2.2 The *Phone Record Format*). The length of the question retrieved must match the phone number size as specified in the phone file. The data must be in valid phone number format. That is, it must have the right number of digits with optional dashes or parentheses around the numbers, (i.e., (415) 777-0470). Here is the syntax and an example for the PHONE,C question type:

```
!PHONE,C,<label or location>

EX:
{PNUM: 1/41.15
Enter new number
!VAR,P}
{!PHONE,C,PNUM }
```

This PHONE,C example replaces the old phone number with the new. The number is checked for a proper area code and prefix. The time zone, state and daylight mode of the number is adjusted if necessary.

There is another setting for the !PHONE, C where you can set the length as well.

```
EX:
!PHONE,C,label,<length>
```

The length is not required, but if it is not set, it defaults to 10.

PHONE,D Sends a number to the dialer to be called. This lets you use autodialers or Getspecific mode with a dialer (not used by predictive dialers, they use PHONE,4). It will allow dialing to numbers other than the one you have the phone record for. Provide a label or location specification where the phone number to call is stored.

```
     !PHONE,D, <label or location>
```

If you are calling a different number, you would most likely use this in conjunction with a PHONE,C to reset the phone number to the number you called in the phone record. Note that this command reloads the phone number from the phone file, which means that any PHONE,P statements prior to the PHONE,D will be lost.

PHONE,E erases the history for the current call (saves the number without a history for this call) even though you gave it a status to move it to some other calling stack. There are no parameters.

PHONE,G gets information from the phone record and puts it in the data record. See 6.2.2 *The Phone Record Format* for phone record locations. In PHONE,G and PHONE,P statements, you can reference the fields using the field name or the data location in the phone file. Here is the syntax and an example for the PHONE,G question type:

```
!PHONE,G,<FROM label or location,width (phone record)>

     EX:
     !PHONE,G,STATECODE
```

This PHONE,G example gets the information from the STATECODE field (1121.2) in the phone record (state code) and puts it into the data location specified on the question label line (or the next available field if none is specified). If a width is used, it may be specified either on the !PHONE line or in the data location field on the question label line.

The PHONE,G statement will check for valid phone positions if a location is used. The number of columns that you're getting (the width) can be from 1-240 per statement in the phone text area, but must be the actual width of the field for non-text area items, except the time zone and call history field. In large phone file systems, references to CfMC variable locations can be "actual" or "standard"; standard is the default and would start in position 5000. Phone,G statements can move 3,000 columns of data between the phone file and the data file.

See the phone parameter "PHONE_REC_LOC_STYLE" above for more information.

The fields can also be referenced by *name inside brackets (eg. [time_on_number#1-60]) in FONEUTIL and SURVSUPR "SELECT" statements (see 6.7.1 *Managing the Phone File with FONEUTIL*). Note that underscores are optional in the field names. In the table below, items are labeled 'S' (String data) or 'N' Numeric data). Numeric fields may be referenced using ranges and numeric equations in addition to comparisons. 'Fields referenced as 'Strings' must have "$" at the end of the name when referred to and must be compared to items in quotes to (eg. [caseid$]="0001"). Other rules apply. See 2.6.2, *Using Data Location* References.

The names can also be used in Survent where the phone file is specifically referenced, such as PHONE,G and PHONE,P statements to get or put data to/from the phone file, and \[phone location] text display references.

PHONE,H this sends the data in the location specified to a PRO-T-S dialer as a 'USER_INFO' message (See 6.8 *AUTOMATIC PHONE DIALERS* for more information). If no data location is specified, it will send "IVR_STATUS ... NOTREADY". The syntax is:

```
!PHONE,H,<label or location>
```

In addition, you can use the DONTHANGUP parameter with the !phone,h, statement. This will keep an interviewer on the phone when another interview is started at that number. !PHONE,H,DONTHANGUP tells the dialer that you are staying on the phone for the next interview, so it should stay connected. This supresses the "HANGUP" and "IVR status" messages to the dialer until the next phone record is sent to this interview. This is only in effect until the interviewer gets another phone number or receives another !PHONE,H,DONTHANGUP command

PHONE,I initializes a phone number, taking the current record and making it look as if it were a fresh number. All history fields are cleared, but other system fields (time on call, calls made) are not. There are no parameters.

PHONE, J is used to change a number from one market to another. You specify the name of the market you want to move the number to in the !phone, j statement.

The syntax for a PHONE,J statement is:

```
    {!PHONE,J,<market name>}
```

In addition, the old market name and the number of changes made to the market is now stored in the phone file; “Prev_market,” which is the previous market number, is in column 5214.3. "Market_changes," the number of times the market has been changed, is in column 5217.3.

PHONE, L is used to call a number in the market specified.

The syntax for the PHONE,L question type is:

```
    <!PHONE,L,<label or location>
```

This is used in conjunction with special interviewer types or specific interviewer IDs to have numbers from a particular market called by particular interviewers. When using PHONE,L, you do not need a prior PHONE,4 or PHONE,5 to get the number.

This works like the !PHONE,5 statement; you give it a two-digit data location, and it returns a code to the data which you use in case the number cannot be retrieved to determine what to do.

There are also two new codes returned: “BM” is returned if a market is asked for and there are no numbers in that market, “M0” is returned if the current market weight is “0” for that market (and no numbers can be returned).

In addition, !PHONE,L enables you to put a list of markets or a file pattern to match a list of markets instead of a single market name.

Where <label or location> can contain up to 80 characters of market name references such as "m*,r*" to get numbers from all markets starting with m or r.

See the list of returned codes under PHONE,5 above.

NOTE: To handle the situation of timed calls coming up from other markets (since they are not under the control of markets), you should pick up timed numbers and immediately reassign them Bucket 9, which will put it back into its proper market and cause it to be called when the market weight is set above 0; make sure you have USE BUCKET 9 set to “yes” if you want to retrieve it today.

PHONE,M sets the market weight. This is usually used to set the market weight to “0” when a quota target has been reached for the market and you no longer want to attempt to get numbers from there. See *6.6.3 Controlling The Sample With Markets* for more information.

The syntax for the PHONE,M question type is:

!PHONE,M,<market name or [location or label],weight>

Market name is the name of the market to apply the weight to. This can be referenced directly or placed in the data and referenced by data location or label name. Weight is a number between 0 and 9. If you set it to 0, the market is no longer called.

Using the “label or location”, you would store the name of the market and then reference it from the data. In this manner, you can update all your market weights with the following few statements instead of one statement for each market:

```
EX:
{mkt: !phone,g,51,4}
{!if quotan([quotanum]) >= quotan([targnum])
!phone,m,[mkt],0}
```

Note that you can reference quota names or numbers in the same fashion.

PHONE,N lets you get numbers that are not allowed by default when using PHONE,5 to get specific numbers. The PHONE,N must be specified before the PHONE,5. The syntax for PHONE, N is:

```
!PHONE,N,type to get
```

It has other parameters:

PHONE,N,HIDDEN allows you to get hidden numbers using a "get specific number" prompt.

PHONE,N,Resolved allows you to get numbers that have been resolved.

PHONE,N,Special allows you to get numbers that usually require a special interviewer type to get. Once a number has been retrieved, it uses the existing information to determine what to do with the number next. For example, if you get a number that has already met the maximum number of calls, it will be returned to the resolved stack with a status 94 when you are done, if you do not get a complete.

NOTE: Once PHONE,N has been executed, it stays in effect for the entire interviewing session. It is not turned off from interview to interview and is not turned off until you exit Survent.

PHONE,O sets the special interviewer type in the phone record. Special interviewer types are used in the case of different languages, interviewers better at closing suspends, etc. This can be a number between 0 (no special type) and 9. It causes the phone number to be sent to a special interviewer stack when it comes up to be called at its scheduled time. It will be offered only to a special interviewer until the special interviewer type is changed to "0" again.

NOTE: This does not immediately put the number in the special interviewer stack, but is put in a stack depending on the value in the PHONE,S statement; when the number comes up to be called, it is moved to the special interviewer stack. To put the number in the special interviewer stack immediately, use PHONE,S,191-199 or PHONE,S,201-209. This will also mark the record as a "Special Interviewer number from that point on (column 22). If you put the number in the stack and do not use the PHONE,O, the following call(s) again will be released to go to any interviewer.

Here is the syntax and an example for the PHONE,O question type:

```
!PHONE,O,interviewer type

EX:
!PHONE,O,1
```

PHONE,P moves information from the data record to the phone text area. PHONE,P is the reverse of PHONE,G, but data may only be written to the phone text area or to columns 25-34 (caseID field) to attach a case ID and change the phone record to Phone Says Datarecord mode. See PHONE,G for details on the phone text area.

The syntax for the PHONE,P question type is:

```
!PHONE,P,TO location(phone text),FROM 
location(data),width
```

The TO location must be a field name or column in the phone text area. The FROM location can be a question name or question number, in which case the width is optional; or a data location, which requires a width. The width can be specified on the PHONE,P line or as part of the data location (i.e., [161.10]). If the data location is a question label, the width will be assumed to be the width of the labeled question unless overridden by a separate width.

```
EX:
!PHONE,P,51,[161],80
```

This PHONE,P moves 80 data columns starting at column 161 to the first 80 columns of the phone text area.

The PHONE,P statement can move up to 3,000 columns of data between the phone file and the data file.

PHONE,R is used to send a phone number to a ROLM (or other) autodialer. The location of the phone number in the data is specified on the PHONE,R statement. You would typically load a phone number with a PHONE,4 and then get the number with a PHONE,G, massage it to fit your autodialer, and then send it to the autodialer with a PHONE,R.

The syntax for the PHONE,R question type is:

```
!PHONE,R,data location
```

PHONE,S specifies the status code for the current phone record when it is put back in the phone file. Only the last status code assigned is saved when the phone record is returned to the phone file. Status codes 1-99 will resolve a number, codes 101-199 and 220-249 will record a history and assign for callback, codes 200-219 will not save a history but will assign for callback. See *6.4.3 Status Codes Returned From SURVENT* for available status codes.

If no status code is assigned, the program will assign status code “1” for a completed interview (case written), “104” for a suspended interview (timed callback), otherwise it will display the standard status screen to the interviewer for them to assign a status. If you back up over a PHONE,S statement, the status is reset to 0.

Here is the syntax and an example for the PHONE,S question type:

```
!PHONE,S,status code

EX:
!PHONE,S,5
```

This PHONE,S will assign status 5 (nonworking number) to this call attempt. See the following section for a more detailed example.

NOTE: Statuses that put the number in special interviewer stacks (191-199, 201-209) also put the special interviewer flag in column 22. This flag stays in effect for future calls unless you either 1) use Phone,O to change it to something else, or 2) You put the number in some other special interviewer stack.

Timed callbacks have an additional “time to call” parameter. These statuses (status 104 or 160-179), allow you to specify a number from 1 to 2 billion as the number of minutes later to call back (1 minute to 20 years from now), or -2 to -60 which means this time tomorrow minus these many minutes). Here is an PHONE,S,104 example specifying to call back 50 minutes from now:

```
EX:
!PHONE,S,104,50
```

You may also specify a number of days from now to call back in. The number of days can be any number from 0 (in the next 24 hours) to 5000.

```
EX:
!PHONE,S,104,5 DAYS
```

This will set the call as an “approximate time” call in this time period 5 days from now. The time (number of minutes) specified can also be in the form of a label or data location.

NOTE: If the data location is not enclosed in brackets and looks like an integer number, it will be treated as a constant, not a data location.

```
EX:
PHONE,S,104,[53.20]
```

If you specify **PROMPTNOW** in the “time to call” field, the program will prompt for the time to call immediately instead of at the end of the interview. This may be useful when using an autodialer, for instance. Be careful not to override the status later in the interview if you have already prompted for a time.

Statuses above 900 signify special features of the PHONE,S.

Specifically, here area a few useful examples of the use PHONE,S:

!PHONE,S,**104** (TIME TO CALL) programmatically provides a date/time to call. Use !PHONE,S,104 or 161-179 statement. The syntax is:

Syntax:

```
{!Phone,S,<104 or 161-179>,<label with date/time string>}
```

Specify date/times with the standard date/time format, such as: YYYYMMDDHHMM. For instance, 200610251525 is for October 25, 2006 at 3:25pm.

!PHONE,S,**156** puts numbers in "GET SPECIFIC ONLY" stack , (stack 338), and they will be held there until you retrieve them specifically.

PHONE,S,**998** indicates that if at the end of the interview the program has not been given a status. It does not present the standard status screen, but instead gets a Survent Blow error #318 and it is left up-in-the-air. This is so you can detect if you have forgotten to set the status programmatically.

If you mistakenly do not set a phone status, by default you get a CfMC status screen from which you can choose one of the standard statuses. But, there are times when you don't want interviewers picking from the list of standard statuses and you want to fix the error not setting a status. To do this, use !PHONE,S,998 at the top of the questionnaire, and if a new status is not set before the end of the interview, the questionnaire will get a blow error that explains that no status was set (and maybe the programmer will fix the error after this is reported a few times).

The blow error you will see is:

```
BLOW #318: Tried to get default fone status screen
after it was disabled
```

Un addition, the software will check to make sure the status you try to use on a !phone,s statement is an available status for users. The statuses that are considered "Bad" are statuses that are reserved for CfMC or a dialer's use. The statuses that will be "bad" are: 70-100, 103, 180-183, 185, 190, 200, 210, 215-219 and 250-255.

If you use the special status **999**, you may then specify a question label or data location to get the status from.

Here is the syntax and an example for the PHONE,S status 999 question type:

```
!PHONE,S,999,<label or data location>
```

```
EX:
!PHONE,S,999,STATUS1
```

PHONE,S,999 supports the PROMPTNOW keyword (above)You can now get a status out of the data using:

```
Syntax:
!Phone,S,999,<label>,PROMPTNOW
```

And, the program will prompt for a time to call immediately instead of at the end of the interview. Note that this feature is already supported for !Phone,S,104,Promptnow and also for statuses 160-179 (all TIMED statuses).

New PHONE,S statuses 252-254 are used to move records to particular stacks. The syntax is:

```
252 move to 'call first' stack
253 move to 'email send' stack
254 move to 'email reminder' stack
```

The "call first" stack is the first place Survent looks for numbers, so those are called before any numbers. The "email send" and "email reminder" stacks are holding stacks used in conjunction with webSurvent.

PHONE,T takes the time specified by the FROM location, converts it to respondent time, and puts it into the data location for this PHONE question.

The date/time in the FROM field must be 12-18 characters of the format YYYYMMDDHHMMSSXJJJ which matches the standard SPC,3 date/time format. If the date/time in this location is bad, then (BAD DATE) is put into the data. The width of the receiving location must be 12-18.

Here is the syntax and an example for the PHONE,T question type:

```
!PHONE,T,FROM location

EX:
{ [2/43.12]
!PHONE,T,[4/10]
```

PHONE,U clears Ownership mode in the phone record and sets the current owner to blanks. There are no parameters (See 6.6.5 *Ownership Mode).*

PHONE,V is used with a dialer to tell Survent what to do if the dialer has sent an interviewer a CONNECT and the phone is off the hook. The default is that a message is displayed to "PICK UP THE PHONE, INCOMING CALL". PHONE,V sets this back to the default after using PHONE,W to change it.

PHONE,W is used with a dialer to tell Survent what to do if the dialer has sent an interviewer a CONNECTed call and the phone is off the hook. If PHONE,W is specified, the dialer hangs the phone up and gives the call a status of 109 to be called back later. If you specify PHONE,V, the program will return to the default, which displays a message to the interviewer to pick up the phone.

PHONE,X marks the interview as one to be validated by a validator; when the interview is finished, the phone record and associated data record are sent to a validator to be reviewed or for additional questions. This works with the SER/EIS dialer or without a dialer, in which case the call has to be forwarded to the validator as well (See 6.6.6 *Validation of Completed Interviews*). A Status of 97 is given to the call after it has been validated.

***PHONE,Z** moves information from the data record to the "call note" area for this call (see PHONE,G call history syntax above). The call note may be up to 66 columns wide. The syntax for the PHONE,Z question type is:

!PHONE,Z,TO location(call note),FROM location(data),width. The TO location is a location in the call note field for this particular call (e.g. Call #5) in the call history area. The total length of the call note may not exceed 66 columns. This may only be used if you have 100-column call history fields.

6.3.2 Other PREPARE Statements Related to the Phone System

In addition to the PHONE statements, there are other commands in PREPARE related to the phone system. See chapters 2 and 3 for more information. Here is a list of the related commands:

HEADER STATEMENT:

FONE_TEXT_LENGTH=# Specifies the number of columns of text in the phone file. The maximum is 4900. This must match the value specified in the phone parameters so that PREPARE can check to make sure references to the phone text area are within bounds.

MODIFY_FONE_FILE=Y/N/P Says whether SURVSUPR can modify phone file parameters using the MPF command. The default is "Y"es. "N" mean no and "P" means a password will be required.

QUESTION TYPES:

!SYS,3,<portnumber>,<baudrate> This talks to a serial port in DOS. Text on the SYS,3 command is sent as commands to the port (eg. ATDT<phonenumber> to dial a number) See example, file MODEM^QPX.

!ZSPC,11,1,[return code],[name of file],[where to put data] This statement is used in conjunction with the PHONE,A statement to read an ASCII sample record and load it into the phone file while a study is live. See PHONE,A above for more information. See example file PHONA^QPX.

QUESTION TEXT:

\[<label or location>] This allows you to display any label or position in the phone file. For example, \[1.10] would display the phone number, as would \[phone_number]. You may have as many of these references in the question text as you would like. A special feature is that if you use \[20001.80] (version 7.7), the suspend text will be displayed.

\^<character> Send a control character. This is used to send commands to a modem or autodialer attached to a terminal in

UNIX. For example, the string "\^RATDT1\|Getnumb|\^M\^T" sends a command to dial a number on a WYSE terminal. See example file AUTOD^QPX.

CONDITIONAL REFERENCES:

The following FUNCTIONS and "Extended" functions are related to the phone system (see also 2.6.4 *Using Functions in Conditional Statements*):

DIALER() This function tells the program whether this interview is being controlled by a dialer. It returns a "1" if predictive mode, "2" if preview mode, and "0" if neither.

FONESTATUS() This returns the current phone status that is set. It is usually used in conjunction with a dialer to see what status the dialer has returned. It will return "0" if no status has been set, "1" if it is a connect from a dialer, otherwise a value from 2-250 depending on the status set by a dialer or a PHONE,S statement.

FONE_TEXT(column,width>) This returns the value in the phone file for the location specified. For instance, FONETEXT(1,10) would return a phone number.

LASTCALL() This function tells you whether this is the "last" call to a number. It returns a "1" or "true" if the current interview is the last attempt or higher and "0" or "false" if not. By "last" call, we mean the call after which the "maximum attempts" requirement in the phone file is met, eg. if Maximum Attempts is set to 10, "!IF lastcall()" would be true on the tenth and any subsequent call.

XF(MARKET_WEIGHT(<marketname or [<label or location>])) This returns the market weight of the name or location of the market referenced.

XF(MAX_ATTEMPTS) This returns the current setting of "Max Attempts" from the phone file.

XF(NUMBER_REDIAL) This returns the number of times the number was re-dialed, either automatically or by using the REDIAL command, during this interviewing session.

XF(STRING_TO_NUMBER ()) turns strings into numbers. This function reads any variable or function and returns it as a numeric value if the result is a number. For instance, it will read "1" and return it as " 1" (a right-justified numeric 1). So, it has a couple of specific uses:

To convert the value of a string function to a number: for instance, XF(STRING_TO_NUMBER(LOCALSCR(5,3)) returns the number in column 5 for a length of 3. Otherwise, the function result is always considered a string.

Since it right-justifies the value first, this function can be used to read a !VAR type variable that is left-justified, and return the value as a number; such as the result of a !VAR,N response.

XF(TIMED_CALL) returns how many numbers are scheduled for a certain time so far. This function can be used if you are trying to schedule calls and want the interviewer to be able to know how many calls have been scheduled at that time. Typically, this would be used in an !expression statement. The syntax is:

XF(TIMED_CALL, <time to get>)

Where <time to get> is a standard CfMC date/time string. The program reports any time you want, and it will return the number of numbers scheduled for that time. Here is an example script reporting in five-minute increments for whatever hour during whatever day you want the information for:

```
{ wd:
which date
!var,,20 }

{ wh:
what hour
!num,,,1,12 }

{ ampm:
 AM or PM
!fld
AM AM
PM PM }

>rep $a=00,05,...,55
 { wdv$a: .5 !expr,,xf(TIMED_CALL, \:wd: \:wh::$a\:ampm: ) }
```

```
>endrep
{

>rep $a=00,05,...,55
\:wd: \:wh::$a\:ampm: calls scheduled: \:wdv$a <br>
>endrep
!disp }
```

XF(TIMED_CALL) will return negative numbers on errors:

-1: no date

-2: no phone header

-3: manana_days not set

-4: error parsing date

-5: unused

-6: date before current time

-7: date beyond manana array size (see fonebuld 'manana_days' command)

-8: current time is after time when call can be scheduled

-9: date is after study shutoff time

Also, see the MANANA_DAYS=<1-60> in FONEBULD, which you will need to use to allow this feature, and SHOW_MANANA In the Supervisor to show the same information (number of calls scheduled by hour for X days).

NOTE: The default for "manana_days" is now 7 days, so you can get a list of at least the next week's calls.

6.3.3 Sample PREPARE Specifications Using PHONE Statements

The following example illustrates:

- Designing your own status screen
- Collecting a callback name
- Changing an existing phone number using PHONE,C
- Using a PHONE,S to assign a status

- Putting comments (text following ' ') in to self-document your specs

```
EX:
>PURGE_SAME
~PREPARE COMPILE
[PHTST,CASE_LENGTH=200,COMMENT="EXAMPLE PHONE STATUS SCREEN QFILE",
&FONE_TEXT_LENGTH=200]
{!PHONE,5} ''Allow optional input of desired # by interviewer
{LOADNUM: ''This gets a phone # from the phone file (no display)
!PHONE,4}
{PNUM: ''This puts phone number into the data file
!PHONE,G,PHONENUMBER,10}
{NAME: ''This puts callback name(if any)into data file
!PHONE,G,101,30} ''Note that position 101-130 in the phone record
''has been set aside for the callback name.
{
\T
!DISP,2} ''Clear the screen
{ ''Put first and last two histories of the number on the screen
!PHONE,2,2,1}
{ ''Put the phone number on the screen for outside
!IF [PNUM.3] <> 415 ''this area code. Uses lines 7-8.
\(7,,8) The number to dial is: 1-(\|PNUM.3|) \|PNUM+3.3| - \|PNUM+6.4|
!DISPLAY,2}
{ ''Put the phone number on the screen for inside
!IF [PNUM.3] = 415 ''this area code. Uses lines 7-8
\(7,,8) The number to dial is: \|PNUM+3.3| - \|PNUM+6.4|
!DISPLAY,2}
{ ''Put callback name (if any) on screen
!IF [NAME^^NB]
\(9,1,9,79) The person to ask for is: \:NAME
!DISPLAY,2}
{STATUS: ''Build status screen starting on line 10
\(10,1,22,79)
Hello, my name is ________ and I am calling from Computers for
Marketing Corporation. Would you be willing to speak to me today
about ________?

REASONS TO CALL BACK        REASONS NOT TO CALL BACK
--------------------        ------------------------
01 No answer                11 Refused
02 Busy                     12 Non-working number
03 Callback                 13 Non-residential number
04 Number change            14 Language problem
05 Answering machine        15 Other phone problem
Press <Enter> if you have someone on the line

!CAT,N
01 No answer
02 Busy
03 Callback
04 Number change
05 Answer Machine
11 Refused
12 Non-working number
13 Non-residential number
14 Language problem
15 Other phone problem
(SKIPTO CONTINUE) ^^ continue}
{CBACKNAM: [NAME] ''If this is a callback, then collect
!IF STATUS(03) AND [NAME^^B] ''callback name
May I please have your name so we can ask for you when we call back?
Enter respondent's name or REFUSED if refuses
!VAR,,30,2}
{STORNAME: ''Store callback name into phone text for future use
!IF STATUS(03)
!PHONE,P,101,[NAME],30}
{GETNWNUM: 11 ''Collect new phone number
!IF STATUS(04)
Please enter the new phone number in the format AAAPPPXXXX with
no spaces or other punctuation.
!VAR,N,10,10}
{NEWNUMB: ''Store new number in the phone file
!IF STATUS(04)
!PHONE,C,GETNWNUM}
{RESET1: ''Reset back to top if you have changed the number
!IF STATUS(04)
!RESET,LOADNUM}
>REPEAT $A=101,102,104,105,106,002,005,006,003,013; &
$B=01,...,05,11,12,...,15; &
$C=NA,BUSY,CB,NC,AM,REF,NONWORK,NONRES,LANG,OTHER
{$C:
!IF STATUS($B) ''Set phone number status if not continue
!PHONE,S,$A}
>END_REPEAT
{ ''Drop case if no interview
!SPC,B}
{CONTINUE:
!PHONE,9} ''Suspend allowed from here on
(continue with questionnaire)
```

The example below uses a PHONE,G to display phone text on the screen.

```
EX:
{INTRO:
Press <Enter> to get a phone number
!VAR}
{ ''get phone number and display status screen
!PHONE,0}
{FONETEXT: [321] ''get phone text from the phone file
!PHONE,G,60,20}
{ ''display the phone text
You are calling: \:FONETEXT:
!DISPLAY,2}
(continue with questionnaire)
```

You could also use a direct data reference or direct phone file reference:

```
You are calling: \|321.20| (data reference)
or
You are calling: \[60.20] (phone file reference)
```

6.4 PHONE SYSTEM OPERATIONS

This section discusses how numbers are scheduled, how Survent determines which phone number to get, and what it does when it is done with a number.

6.4.1 How the Phone System Schedules Calls

The phone system tracks numbers according to how they are scheduled to be called. We will refer to a group of numbers with the same call scheduling state as a stack. Think of this as a stack of paper, with each phone number being a piece of paper and sets of these placed on top of each other. When Survent needs a phone number, it uses a formula (explained in *6.4.2 How SURVENT Chooses A Phone Number*) to determine which stack should be used. It then takes the top number off of that stack and presents it to an interviewer. When the interviewer finishes that call, the phone number is returned to the phone file with a status (busy, no answer, complete, refusal, etc.). The phone system uses this status to determine which stack to put this number at the bottom (or top) of.

There are three types of stacks:

1. Specific timed call stacks
2. System scheduled call stacks
3. Special purpose stacks.

Each phone number records the current stack by number and by name. One or the other can be referenced in FONEUTIL (see *6.7.1 Managing the Phone File with FONEUTIL*) and in producing reports where you want to categorize by stack. You can also see the previous stack number and the number of the stack last hidden from.

SPECIFIC TIMED CALL STACKS (1-313)

Specific timed calls use stacks 1-313. These are calls where the interviewer has entered a callback time to the respondent or there was a preset time to call in the original raw phone number record.

Stacks **1-288** hold calls scheduled for a specific time today. Each of these stacks holds calls scheduled for a 5-minute period (12 per hour multiplied by 24 hours = 288 stacks). Stack 1 is for calls scheduled from 12:00am to 12:04am, stack 2 from 12:05am to 12:09am, etc. If you scheduled a call to be made at 2:32pm today, the number will be put in the stack that holds all timed calls between 2:30pm and 2:34pm.

Stacks **289-312** are used to call numbers within a certain hour (approximate time calling). 289 is for Midnight until 1 am, 290 is for 1am to 2 am, etc.

Stack **313** holds all the timed calls scheduled to be called later than today. Whenever the current time reaches the time of a number in stack 313, the program moves all the numbers that are still in stack 313 (from yesterday), but are now scheduled for today, to the appropriate stack between 1 and 312. ***Note:*** This integration of numbers takes place when the first number is asked for on each day, or can be done in FONEUTIL using option I (Integrate).

SYSTEM-SCHEDULED CALL STACKS (350-9650)

System-scheduled calls (such as new numbers and no answers) are stored in stacks 350 to 9999. The number of stacks used depends on how many different time zones and market areas you are calling. There are 10 stacks for each time zone, and up to 8 time zones. Each market contains all stacks in all the time zones. Each relative stack has the same meaning for its time zone. The stacks are assigned in low to high time zones. Here are the stacks for the first time zone specified:

Stack Numbers and Descriptions

350 New numbers

351 Call in day part time 1 (respondent time). Calls that will only be made for day part time 1 interviews are stored here.

352 Call in day part time 2 only.

353 Call in day part time 3 only.

354 Call day part time 1 or time 2. Calls that allow either DPT1 or DPT2 interviewing are stored here.

355 Call day part time 1 or time 3.

356 Call day part time 2 or time 3.

357 Call day part time 1, time 2, or time 3. Calls that can be made during any part of the day are stored here.

358 All targeted attempts made. Calls that have had all targeted attempts made for each of the available day parts, but have not made the maximum number of calls are stored here.

359 Bucket 9, a place to save numbers to be released later for this time zone.

350 New numbers

351 Call in day part time 1 (respondent time). Calls that will only be made for day part time 1 interviews are stored here.

352 Call in day part time 2 only.

353 Call in day part time 3 only.

Subsequent time zones use 10 stacks each, starting at 360. ***Note:*** Time zones do not need to be consecutive; for instance you may use time zones 5,6,7,8,10,11,23,24.

In the following table each cell represents that time zone's corresponding call stack for different call conditions.

1st	2nd	3rd	4th	5th	6th	7th	8th	Description
350	360	370	380	390	400	410	420	Fresh numbers
351	361	371	381	391	401	411	421	Call in day part 1
352	362	372	382	392	402	412	422	Call in day part 2
353	363	373	383	393	403	413	423	Call in day part 3
354	364	374	384	394	404	414	424	Call in day part 1 or 2
355	365	375	385	395	405	415	425	Call in day part 1 or 3
356	366	376	386	396	406	416	426	Call in day part 2 or 3
357	367	377	387	397	407	417	427	Call in day part 1, 2, 3
358	368	378	388	398	408	418	428	All targeted attempts
359	369	379	389	399	409	419	429	Bucket 9

If you use markets, the stacks change. The first market by the first time zone uses stacks 350- 359, and the second time zone for that first market is 360-369, the third is 370-379, etc. until all of the time zones for the first market have been defined. The next set of 10 stack numbers will be used to assign the first time zone for the second market. The next 10 will be for the second time zone for the second market and so on.

The maximum stack number is 9,650, and the maximum market is 255, so you may use 255 markets with 1-3 time zones, 241 markets with 4 time zones, 193 with 5 time zones, 160 with 6 time zones, 137 with 7 time zones, and 120 with 8 time zones (See 6.6.3 *Controlling The Sample With Markets*).

SPECIAL-PURPOSE STACKS (314-349)

The special purpose stacks are numbered 314-332.

Stack **314** contains resolved phone numbers. This includes completed interviews, non-working numbers, terminations, refusals, maximum number of attempts, etc. Any number that is resolved will not be presented to be called again.

Stack **315** is the error stack. Problem numbers (a new or changed number given during the interview and its time zone is unknown) get put here. They will never be presented for dialing, will never move to another stack, and don't show up on displays of the active stacks. If a record is put in the error stack while the study is live, it will keep all of it's information as it was prior to being moved, but the "error stack" flag will be set. This allows us to better track what happened to the phone number. When you delete the number and add it back in using FONEBULD, the "error stack" flag will be ignored.

Stack **316** is for duplicate phone numbers found in FONEBULD when DUPLICATES OK is not set to YES. They will never be presented for dialing, will never move to another stack, and don't show up on displays of the active stacks (e.g., FONEUTIL's P option).

Stack **317** is for numbers that are hidden with FONEUTIL or SURVSUPR. These numbers will not be accessible until they have been revealed, where they will return to the stack they were in before being hidden.

Stacks **318-326** are for numbers to be called back by a special type interviewer, an interviewer who may have a special skill, such as one who speaks a foreign language.

When numbers are assigned to a special interviewer (using the PHONE,O statement or assigning it using column 22 of the raw phone record) they are initially put in one of the of the raw phone record) they are initially put in one of the regular stacks. When that number is presented to Survent, it is immediately rerouted to the special interviewer stack. If an interviewer of that special type is available, the number will be presented to them; otherwise it is presented as soon as the special interviewer is available (and prior special interviewer numbers have been called). Within the special interviewer stacks, numbers that come from timed stacks have priority. Stack 318 for special interviewer type 1. Stacks 319-326 are for special interviewer types 2-9, respectively.

Stack **327** is for the owned records stack. This is where numbers that are scheduled to go to specific interviewers that are not currently signed on are put.

Stack **328** is for “on-the-floor” records. These are numbers that were up in the air (in use by an interviewer and never returned to the phone file), but a FONEUTIL V option put them here.

Stack **329** is the place where numbers for the “dummy” interviewer are stored (a “dummy” interviewer is a Survent session that handles numbers returned from a dialer before they are sent to interviewers for calling). The ttyfile parameter “DIALER:EIS,DUMMY” allows the SER/EIS dialer to pass unconnected numbers to an interviewer station logged on with special interviewer type 9 to handle dispositioning of these numbers through Survent. This allows you to write special code based on prior calls, etc., to handle the statusing of disconnected numbers instead of using the standard statuses provided my the SER/EIS dialer.

NOTE: You can logon to as many stations as you wish with special interviewer type 9 to handle the server statusing activity.

Stack **330** is where FONEBULD puts numbers that matched records in the “forbidden” file. This file is used to keep studies from calling numbers of people who asked never to be called, were called previously, etc.

Stack **331** is where numbers that have an unsupported time zone or other problem when interviewers try to change the phone number. This usually occurs when a PHONE,C is used to change a phone number and the changed number is in a time zone that is not supported by the current phone file. Numbers in stack 331 put “BADZONE” in the stack name field (1161/5161.15). (For more on this, refer to the !PHONE,C statement earlier in this chapter.)

Stack **332** is where numbers are placed when they are sent to a predictive dialer. Numbers in stack 332 get marked as “ATDIALER” in the stack name field. If a number is connected to an interviewer it is moved from the atdialer stack and marked as "uita" (upintheair). If a number comes back with some other status it is moved from the "atdialer" stack to the appropriate stack based on the call result from the dialer. When the study is shut down, numbers are returned from the atdialer stack to the top of the previous stack they came from whether or not the dialer sends them back.

Stacks **333-348** are reserved for future use.

Stack **349** is for “free” or deleted records. They are records that were deleted from the file using FONEUTIL’s “D” option. These records are not generally accessible by any of the programs, except if you use FONEUTIL’s USEFREESTACK option, which allows the program to read and/or convert the records to be readded to a phone file. Each number occupies a physical slot in the phone file. When a number is deleted in FONEUTIL, the slot for that number becomes free.

FONEBULD, when adding numbers to a phone file, first fills any free slots before appending to the file.

6.4.2 How Survent Chooses a Phone Number

Here is the inquiry order when Survent gets a phone number with a PHONE statement:

1 Is the interviewer asking for a particular number?

If so, look for that number. If it is available, get it. If it is resolved or owned by a special interviewer type, only get it if the proper PHONE,N option is used.

2 Is the interviewer a special type?

If so, look in the special interviewer type stack. If the interviewer has more than one special type, start looking in type 1, then type 2, etc. Timed callbacks that are due to be called will be at the top on the special interviewer call stacks.

3 Is Owner mode on?

If so, check the owned records stack for any records this interviewer may own. (See *6.6.5 Ownership Mode* for more information on this.)

4 Are there any specific timed calls scheduled for now?

If so, use the first one available (regardless of time zone). Are there any approximate time calls for this hour?

5 What time zone will be used for system scheduled calls?

If the TIME ZONE WEIGHTS are equal, the zones are picked randomly in proportion to the number of phone numbers they currently have available. For example, if at 3:00 pm EST there are 1000 numbers available in the Eastern time zone and 100 numbers in the Pacific time zone, then the Eastern numbers will be used approximately ten times as often as Pacific time zone numbers. If weights are applied, they change the proportion for each time zone by its weight.

6 What markets will be used for system-scheduled calls?

If the market weights are equal, get a number from any market, in the same fashion as for time zones. If a market weight is zero, no numbers will be retrieved from that market. (See *6.6.3 Controlling the Sample with Markets* for more information.)

7 Are there any system-scheduled callbacks available for the current day part?

If the “bucket_order” parameter is used, Survent searches the system’s scheduled buckets for the time zone chosen in the order specified. If not, it looks for numbers in the “this day part only” stack first, then the “this day part or one other” second and then the “any day part” stack third, in time zone order. If no system scheduled callbacks are found in any time zones then it looks for new numbers in time zone order.

If NEW NUMBERS FIRST is set to YES in the phone file (either by having been set by FONEBULD’s SHOPFILE command or changed in FONEUTIL or SURVSUPR), it changes the above search order so that it looks at new numbers first.

If a specific time-scheduled call isn’t attempted within MAX TIMED CALLBACK AGE minutes past its scheduled time, it is rescheduled for the same time the next day, unless the MAX TIMED CALLBACK AGE is a negative number, in which case it is resolved with a status 95.

Timed calls are not subject to ZONE WEIGHTS or MARKET WEIGHTS. In general, since timed callbacks are first priority, they will seldom reach the max callback age (the amount of time to allow before rescheduling for tomorrow) without being attempted.

For system-scheduled calls, the four-day part times are used to separate days into morning, afternoon, and evening. The DAY PART TARGETS parameter is used to determine how many calls you want in each day part, unless overridden by ‘preferred time of day’ sampling. Here is an explanation of how system-scheduled calls are handled:

Initially, a number is assigned as a new number in its time zone by FONEBULD unless specifically placed in a call stack with BUCKET=, STACK=, or a time to call in the CALLBACK_TIME field (columns 31-45).

Each call during a day part decrements the number of calls allowed to that day part by that phone number by one.

If the DAY PART TARGETS are set to 1 1 2 (which means 1 morning attempt, 1 afternoon attempt, and 2 evening or weekend attempts), the first attempt can be made at any part of the day since all day parts are available. If the first attempt is made in the evening and is a no answer, the program will still put the number into the "call morning, afternoon, or evening" stack, because it still has one attempt available for each day part. Suppose the second attempt is made in the afternoon, and it is also a no answer. It will be returned to the "call morning or evening" stack, because it has exhausted all of its afternoon attempts.

If the third call is made in the evening and again is a no answer, it will be returned to the "call morning only" stack.

When the fourth call is made and if it is also a no answer, it will either be resolved (total call attempts made), or it will be put in the "All Targeted Attempts Made" stack, where it may be released to be called at a later time. The maximum call attempts will determine where it goes.

Once a number is in the "All Targeted Attempts Made" stack, it cannot be called again until the option to release these numbers has been specified. SURVSUPR or FONEUTIL can be used to release the ATA numbers. This is used to make additional attempts to a number at the end of a study when you may be short on sample.

System-scheduled calls are subject to ZONE WEIGHTS. If you set ZONE WEIGHTS=0 for a time zone, no system scheduled calls will be available for that time zone. ***NOTE:*** Use ZONE WEIGHTS with caution. By overweighting a specific time zone, you may be introducing sampling errors by causing the times people were called to be skewed.

If preferred time-of-day sampling is used, you will specify which day parts you want each call to be scheduled for after each succeeding No Answer in each phone record.

Timed calls (busies, specific time) do not count in the number of day part attempts, even though they occur during them. Busies are considered extensions of the initial call; timed callbacks are given the same consideration.

6.4.3 Status Codes Returned from Survent

Each time Survent gets a phone number, it will return the number to the phone file with a status code for that call. This status code determines which stack the phone number will be returned to, and gets stored along with the history for that call. Additionally, each phone number is assigned a final status. This is 0 while the number is active, and some resolved status when the number is no longer active.

Status codes are also used for tracking the numbers called by the interviewing team, and to produce interviewer productivity reports. They are divided into two groups, resolved status codes and callback status codes.

- Resolved status codes are 1-100, and are assigned to numbers that will not be called back.
- Callback status codes are 101-250.

If no previous status has been set, then Survent assigns a status of 1 for a completed interview. Resolved status 94 is assigned to numbers that have made the maximum number of calls without completing the interview. All other status codes are assigned by the system (in case of errors), the default status screen (see below), or by using the PHONE,S question to specifically assign a status. There are many status codes available to you for your own purposes. For your own purposes, use only the status codes noted as being available to you: 13-69, 108, 110-156, 160-179 and 220-249 (see following pages for a list of the status codes and their meanings). Using system assigned status codes for other than their intended purpose may cause the system to handle the number differently than you expected.

Here is the default status screen used by the PHONE,0 statement. The status codes that the system assigns are listed in parentheses. This default screen will also appear if an interview is aborted with an SPC,B or SPC,H and no prior status code has been set.

```
Reasons to Call Back     Reasons Not to Call Back
A-->No Answer (101)      D -->Refused to start (2)
B-->Busy (102)           E -->Language (3)
```

```
C-->Call Back (104)      F -->Terminated interview (4)
                         G -->Not a working number (5)
                         H -->Not residential (6)
                         I -->Not business (7)
                         J -->Ineligible A/Stop Interview (8)
                         K -->Ineligible B/Stop Interview (9)
                         L -->Ineligible C/Stop Interview (10)
                         M -->Ineligible D/Stop Interview (11)
                         N -->Ineligible E/Stop Interview (12)
```

Most clients design their own status screen, or use the one in the example file PHONE^QPX. To design your own status screen:

- use a single-response CATEGORY or FIELD question for the screen display use PHONE,S type questions to send the status code based on the response to the CAT or FLD question.

To give a status without completing the interview on this call (i.e., assign a callback status): use an SPC,B or SPC,H statement to abort the interview after assigning the status. (See *6.3.3 Sample Prepare Specifications using PHONE Statements* for examples.)

Here is a list of the status codes and their meanings:

0 - No status yet given

Status codes 1-100 are resolved numbers:

1 - Completed interview (Default if interview completed and no status set)

2 - “Refused to start” on default status screen, otherwise usable

3 - “Language problem” on default status screen, otherwise usable

4 - “Terminated during interview” on default status screen, otherwise usable

5 - “Non-working number”on default status screen, otherwise usable

6- “Non-residential number” on default status screen, otherwise usable

7 - “Non-business number” on default status screen, otherwise usable

8 - “Terminate, reason A” on default status screen, otherwise usable

9 - “Terminate, reason B” on default status screen, otherwise usable

10 - “Terminate, reason C” on default status screen, otherwise usable

11 - “Terminate, reason D” on default status screen, otherwise usable

12 - “Terminate, reason E” on default status screen, otherwise usable

13-68 - Available for user-defined resolved status code

69 - Dialer - connected # to interviewer (Can be controlled by “dialer_connect_status: ##” in the parmfile.)

Status codes 70-100,103 are used by CfMC and should not be assigned in a questionnaire:

70 - New number resolved by Fonebuld or Survent if number in “Do Not Call Prefix” file - Don’t call

71 - Dialer – number not dialed

72 - Dialer - specific timed call not dialed

73 - Dialer - unknown number

74 - Saved for system defined status

75 - Dialer - not in service

76 - Dialer - call not completed (can be controlled by “call_incomplete_status: ##” in the parmfile)

77 - Dialer - bad number

78 - Dialer - modem answered (can be controlled by “call_modem_status: ##” in the parmfile)

79 - Dialer - disconnected

80 - Dialer - forced resolved

81 - Number killed in FONEUTIL

82 - Dialer - phone number changed

83 - Bad record

84 - Unknown record

85 - Timed call with no valid callback time

86 - Putfone with record number=0

87 - Bad stuff in phone record

88 - Unknown status

89 - Bad special interviewer type value (not 0-9)

90 - No stack found

91 - Duplicate phone number (from FONEBULD)

92 - No next bucket/stack

93 - Bad phone number

94 - Maximum calls met (final status only)

95 - Timed number too old to call (if maxcallbackage=-###)

96 - Resolved by Fonebuld or Survent if number in “Do Not Call Prefix” file - Don’t call

97 - Reviewed by Validator (see 6.6.6, section on Validation)

98-100 - Saved for system-defined final status codes

Status codes 101-200 are no-answer numbers to call back, unless otherwise indicated:

101 - No answer

102 - Busy (timed callback using BUSY RESCHEDULE) or Dialer

103 - Busy changed to No Answer (dependent on NUM_BUSY_TO_NA)

104 - Call back at specific time (by interviewer or programmatically) or TIME TO CALL

105 - Call back later, but interviewer didn’t give a time

106 - Dialer - nuisance call

107 - Dialer - answering machine

108 - Dialer - “G” status (SER/EIS dialer only)

109 - Dialer – “Interviewer hung up phone” (see !PHONE,W)

110-155 - User-defined No answer callback status codes

156 - “Get specific only” put in stack 338

157 - “Trunk busy”, treated like a status 102/busy. Can be controlled by “call_trunk_busy_status: ##” in the parmfile

158 - User-defined busy. Treated like status 102.(Pays attention to the “number of busies to treat like a now answer” parameter and gets called back in “busyreschedule” minutes.

159 - User-defined busy. Treated like status 102.(Pays attention to the “number of busies to treat like a now answer” parameter and gets called back in “busyreschedule” minutes.

160-179 - Times callback statuses (treated like 104)

180 - Got a Survent blow. Saved in blow stack 339. Use “zap,last” in foneutil to return numbers to the previouscalling state. They will be returned to the “on the floor” stack when you use foneutil’s “verify” or “fix” options.

181 - Dialer - got a connect, then abort

182 - BUSY (timed) changed to NO ANSWER because the busy reschedule would result in a time outside the shop open/shut times, or because RELEASE_TIMED_CALLS is set to Yes and numbers would be called again immediately.

183 - Dialer – “no ringback” status (PRO-T-S dialer) This is a “no answer” type callback status, returned as the result of a successful dial with a dial tone, but the result of the dial is 40 seconds of continuous silence or 6 seconds of continuous noise. No ring backs can happen on a particular call at any particular time and never occur on that call again. If a person ws dialing manually (on any system), they would just hang up and dial again. Typically, a number will not generate a no ring back on two consecutive calls, but there is no set pattern as to when they will occur.

184 - Like status 104, but this attempt is the “maximum_ attempts” attempt, and is is resolved with status 94 instead of rescheduling the number. This overrides the default that continues calling specific-timed numbers. Phone parameter “timed_use_max_attempts: yes” will do the same thing

185 - Dialer – Incomplete callback (SER/EIS dialer only). This is a “no-answer” type callback status.

*186 - Puts the number in front of the stack it came from and history is saved

*187 - Similar to 186, but the number is put in the back of the stack and history is saved.

188 - Puts the number into the hidden stack and the history is saved

189 - Puts the number into bucket 9 and the history is saved

190 - “Sighup” signal received, terminal disconnected, interview suspended

191-199 - Special interviewer stacks for types 1-9 (history kept of attempt)

No call history saved for 201-218:

200 - Not used

201-209 - Special interviewer stacks for types 1-9 (no history kept)

210 - Saved for system-defined callback code

Statuses 211-214 correspond to statuses 186-189, except no history is saved:

*211 - Put number back where it came from (front of stack). If Minimum Age has not expired, put it at the end of the stack.

*212 - Put number back where it came from (back of stack)

213 - Put number directly into hidden stack.

214 - Put number directly into Bucket 9 holding stack.

215 - Put number in front of Bucket 9 stack

219 - Put number in front of Bucket 9 stack, save the history

298 - Forces a “blow” if you mistakenly do not set a phone status instead of the detault status screen. This means you will get status 180 with a blow message that explains that no status was set (Maybe the programmer will fix this after it is reported a few times.). The blow error you will see is:

```
BLOW #318: Tried to get default fone status screen
```

```
     after it was disabled.
```

999 - Assign whatever status is ub tge data field pointed to by !phone,s,999 label.

NOTE: All callback statuses are treated as system scheduled callbacks except status codes

102,104 and 160-179 that are treated as timed calls.

*Marketweightzerostatus of these statuses is disallowed because they can cause the program to get into a loop picking up the same record over and over again.

6.5 INTERVIEWING WITH PHONE SYSTEM

To start a study which has PHONE statements, the phone file must be built and in the proper directory. It should be where the CFMCFONE system variable points. Also, the phone records must have the proper study code, so you cannot rename a phone file from its built study name to another name and start interviewing with it. If you need to change the name, you must convert the records out using FONEUTIL and rebuild the file with the new name using FONEBULD.

If the phone file is not "integrated" yet, the CfMC SERVER will integrate the file (move next day's calls to today's calling stacks) for you, depending on the value of the FORCE_INTEGRATE: server parameter in the PARMFILE (in the CONTROL directory under CFMC). If you get an error message, you will have to integrate the file with FONEUTIL before starting. By default, the server will integrate if there are up to 200 numbers in the "call later" stack, otherwise it will force you to run FONEUTIL to integrate the file offline.

Generally, when using the phone system with Survent, you will want to write your own status screen for more control (see 6.3.3 *Sample PREPARE Specifications Using PHONE Statements*). If you do not have your own status descriptions, you may use CfMC's standard status display question with the PHONE,0 statement.

When Survent executes a PHONE,0 statement, it displays the number, along with its history, and a default status screen.

Here is an example of a PHONE,0 screen display:

```
THE NUMBER TO CALL IS: 312-428-0393 (6) 17:30
HISTORY:
Call # 1: THU AUG 27 1992 15:04 - 5:04 ID:DANN STATUS:Busy
Call # 2: THU AUG 27 1992 15:34 - 5:35 ID:MARY STATUS:Callback

Reasons to Call Back:   Reasons Not to Call Back:
A-->No Answer           D-->Refused to start
B-->Busy                E-->Language
C-->Call Back           F-->Terminated interview
                        G-->Not a working number
                        H-->Not residential
                        I-->Not business
                        J-->Ineligible A/Stop Interview
                        K-->Ineligible B/Stop Interview
                        L-->Ineligible C/Stop Interview
                        M-->Ineligible D/Stop Interview
                        N-->Ineligible E/Stop Interview
```

This screen gives the number, its time zone (6), and the time the number was due to be called back (17:30, or 5:30 pm). The history line gives the date, start and stop times of all calls, the interviewer's ID, status of the call, and number of calls made to this number so far. This is followed by the status screen (see *6.4.3 Status Codes Returned From SURVENT)*.

If you have reached the prospective respondent and wish to proceed with the interview, press **Enter**. If you are not going to conduct the interview, you must enter the appropriate response to either provide for a callback or end calls to that number. The status screen will also appear if you end an interview by responding ABORT to any question {!ALLOW_ABORT} must be set), or there is an SPC,B or SPC,H.

To arrange a specific callback time, enter "C" and you will receive the following prompt:

```
WHEN TO CALL BACK (415-777-0470)?-->
```

This is the same prompt you will see when you set a status using PHONE,S,104 or PHONE,S,<160-179>.

Note: You may back out of this prompt if you need to, unlike the situation where you reach this prompt by entering SUSPEND and cannot backup.

There are many different formats that may be used when entering the date/time. You may enter a specific date and/or time or an "approximate" date/time. Here is a list of the allowable specific date formats:

	Format type	**Examples**
Date:	Month day	Dec 1 or December 1
	Day Month	1 Dec or 1 December
	M/D/Y	12/3 12/3/06 12/3/2006
	DDMMMYY	03DEC 3DEC 03DEC06 03DEC2006
	YYYYMMDD	20061203
	MM-DD-YY	12-03-06
	DD-MM-YY	03-12-06
	MM/DD/YY	12/03/06
	DD/MM/YY	03/12/06
Day:	DDD or DAY	TUE or Tuesday TODAY TOMORROW

To enter dates in the format of D/M/Y or DDMMYY like it is common to do in Europe, put DATE_FORMAT:DDMMYY in your PARMFILE. *See 4.4.4 INTERVIEWING CONFIGURATIONS* for more information.

Here are the time formats:

	Format type	**Examples**
Time:	H:M	9:00 (assumes the hext open a.m./p.m. time)
	H:M <AM/PM>	9:00 PM
	H <AM/PM>	9 PM
	HHMM	2100 (military time)
	+MMM mins	+120 or + 120 mins (120 minutes from now)
	+### HouRs	+10 hrs (10 hours from now)
	+### DAYS	+2 days (2 days from now - approximate time)
	+ WeeKs	+1 wk (1 week from ow - approximate time)
	+### MONTHs	+1 month (1 month from now - approximate time)
	+H:MM	+ 2:30 (2 hours, 30 minutes from now)

Either date (or day) or time may be entered first; they must be separated by a space. A time entered without a date means the next time it is that time. A date entered without a time means at the current time on the date I'm giving you. A time without AM or PM defaults to the next time the shop is open, eg. If it is 6:00pm, if I say "9:00" it will be for 9PM unless the shop is closed then. A day of the week may be entered instead of a date. Enter the 3-letter abbreviation (MON, TUE, WED, THU, FRI, SAT, SUN) or spell out the day fully (MONDAY, etc.).

The month may be specified as 12/10 or December 10 or Dec 10 (using a three-letter abbreviation).

Let's say today is Monday, April 20th. If the respondent says call me on the 24th at 2 o'clock, interviewer enters:

```
4/24 2:00PM
```

If the respondent says call me tomorrow at the same time, the interviewer enters:

```
4/21 or +24:00
```

If the respondent says call me later today at 3 o'clock, the interviewer enters:

3:00 or 3:00PM or 1500

If the respondent says call me back in 20 minutes, the interviewer enters:

```
+20 or +20 mins
```

If the respondent says call me back in 3 hours, the interviewer enters:

```
+3 hrs or +180 or +3:00
```

If the respondent says call me next Tuesday at 5 p.m., the interviewer enters:

```
TUE 5 PM
```

You can also respond to this prompt with a date and time string (yyyymmddhhmm), such as:

```
200104231530
```

This means to call back on April 23, 2001 at 3:30 pm.

When arranging a callback time, ALWAYS enter the time given by the respondent. If the interviewer’s time zone is different from the respondent’s time zone, the callback time will automatically be translated into the interviewer’s time. Survent will also make any necessary adjustments for Daylight Savings Time differences.

As an example, suppose a Pacific time zone interviewer called an Eastern time zone respondent at 9 AM, Pacific Time, on December 4th. If the respondent requested a 1 PM callback, the following translation would be performed by the system:

```
      EX:
WHEN TO CALL BACK (415-777-0470)? --> 1:00 PM
CALL BACK AT: DEC 4, 2000, 10:00 AM (TIME HERE)
DEC 4, 2000, 1:00 PM (TIME THERE)
THIS OK? (Y OR N)?--> Y
```

Callback times are checked for validity by the program. They must be between the OPEN times and CLOSED times for the day and prior to the date/time specified as the "Last Scheduled" time.

If you specify a time before Day Part Time 1 and after Day Part Time 4, you will get a warning message. Once you have entered the time, you will be asked to verify it. If you enter a time that is not allowed, an error message will appear and you will have to choose another time. Press "N" to re-enter the time, or press **Enter** if you wish to have the call automatically rescheduled. Once you have arranged the callback, the number will be available to call at the scheduled time.

If you enter a time without specifying AM or PM, Survent will look forward for the next time that the shop is open at that time (AM or PM) and ask if that is what you want. If it is December and you say to call in January, the next year is assumed. If you specify a time in the format of +30 or +30 mins, the time specified must be at least 10 minutes in the future unless you are in Debug mode.

NOTE: If you get to the callback prompt by mistake, you can enter any of the allowed interviewer commands (^, TERMINATE, etc.) at the prompt, as long as you have used the PROMPTNOW option on the PHONE,S statement that requested a time. Otherwise, you MUST indicate a time to call and status the number because the prompt is displayed after the interview is over.

APPROXIMATE TIME CALLING

You can also set a call to be made some time 'about' a particular hour of the day, or at 'some' time in the morning, afternoon, or evening. The program will pick an hour to call, and the call will have the next priority behind specifictimed calls. There are 24 stacks (289-312) that are 'call at this hour' stacks. If you say to call 'tomorrow afternoon', the program will pick a random stack from 12pm to 5pm tomorrow. If you say “about 3,” it will pick the 2pm or 3pm stack. In all other ways they will act like timed calls.

When the hour comes along, whenever specific-timed calls are all distributed, the system will attempt to call these numbers. You may specify any date format in conjunction with an approximate time format. The list of supported ‘approximate time’ keywords at the "WHEN TO CALL-->" prompt is as follows:

LATER/TODAY/LATER TODAY Some time later today

ABOUT / APPROXIMATELY / APPROX <time>

BEFORE / AFTER <time>

MORNING / MORN From 8am to 11:59am

AFTERNOON / AFT From 12 noon to 4:59pm

EVENING / EVE / TONIGHT / NIGHT From 5pm to 9:59pm

NOTE: MORNING, AFTERNOON, and EVENING will be further restricted by your daypart times (i.e., must be between DAYPART1 and DAYPART4) and also between your OPEN and CLOSED times. For example, if your shop is shut at 5pm, the system will not let the interviewer schedule a callback at NIGHT.

The interviewer will see a note on the screen that the call will be made approximately at a specific time. This allows you to make a distinction between "hard" appointments and "soft" appointments. The “minute” part of the note always reflects the current minute, the call will actually be available any time after the hour prompted with.

```
EX:
7/15 before 3pm
TUE NIGHT
TOMORROW MORNING
```

6.6 ADDITIONAL PHONE SYSTEM FEATURES

The CfMC phone system has many features to control calling. Here are some of the additional features you can use. Most of these features can be used concurrently, for instance, you can do "Preferred time of day sampling" and use "Market control" on the same study.

6.6.1 Retrieving Phone Records Directly Using a Phone Number or Name

Using the PHONE,5 command, phone numbers can be retrieved specifically by phone number or some other criteria. In order to have fast access to specific phone records an index is built, allowing the system to get a specific number without having to read sequentially through the file.

Regardless of the size of the phone file, a number may be specifically retrieved with little strain on the system.

You may build up to three different indices for a study, one of which is always the phone number itself, and up to two more which would allow you to access the phone numbers via some other criteria. In this way you could retrieve a phone record by a name, address, or some special code stored in the phone sample file. Compile and run the example files FNIND^FBL and FNIND^RAW (FONEBULD) and FNIND^QPX (PREPARE) to see exactly how this operates.

CREATING THE INDICES IN FONEBULD

When building a phone file, by default you automatically get an index built for the phone numbers themselves. We assume that you may want to be able to get specific numbers at some time during the study. This index gets built with the extension FNX. This is the file the program reads to find a specific number to get from the FON file by phone number. Phone numbers must be at least 5 digits long.

You may have two additional index files built by saying 'INDEX <column.width>' before building the file, where <column.width> is the position in the phone record which you will be using for the index. The <column.width> must be between 5 and 20 columns long. In addition to the original records processed, any records that are added to the file later will update the index files with their indices. Make sure you say “TEXT_LENGTH=###” before specifying the index so the program knows where you allowed the value to be put.

Each additional index will create an additional index file. The first additional index will create a file with an extension of FNY, and the second will have an extension of FNZ. When the indexes are searched, they are searched in the order of FNY, then FNZ, so put the index you expect to use the most first.

RETRIEVING NUMBERS BY NAME OR ADDRESS IN SURVENT

If you wish to access additional indices, you will need to use the form of the PHONE,5 question that gets its information from a prior question, i.e. "{!PHONE,5,,,info}", where 'info' is the name of the question where the interviewer enters the index information (see 6.3.1 The PHONE Statement, Phone,Letter Subtype G).

The program will first determine whether that data is a phone number, and if so, look in the phone number (FNX) index. If it is not a phone number, the program goes to the next index (FNY), and possibly the FNZ index until it finds a matching code.

If no matching code is found, an error message is returned, and you will be reprompted for a phone number. If you simply press <Enter>, it will continue with the next available phone number.

6.6.2 The ‘Do Not Contact File’

The Do Not Contact file is a file that contains numbers that are checked whenever you add numbers to a sample file either using the Fonebuld program or when Survent adds a new number. It is an indexed CfMC TR file with phone numbers assigned as the case id for the file.

To set this up so that you always check against the file when adding sample, in the CfMC PARMFILE (located in the CONTROL directory) add the line:

```
     EX:
DO_NOT_CONTACT_FILE: /usr/cfmc/control/badrecs.tr (UNIX)
DO_NOT_CONTACT_FILE: f:\cfmc\control\badrecs.tr (DOS)
```

The file (eg. badrecs.tr) can be located anywhere, as long as you have read access to the file. If a telephone number is present in the do not contact file, the number will not be added to the file. To create or append to a dnc file, you can use MENTOR with the following commands:

```
     EX:
~input <ascii filename> ascii=100 id=1.10
~output <do not contact filename>trfiledirectory=<#####>
maybecreate writenow
~end
```

The "####" is the maximum number of records you expect the file to hold. Survent's command to add a number to the phone file (!PHONE,A) checks the file before adding the number, in addition to Fonebuld. Any number that can not be called is added to the fonefile in stack 330, the "never-call" stack, and given resolved status 96. Survent's !PHONE,A command will use the dnc file specified in the parmfile, if there is one.

Fonebuld can override the parmfile specification by having a command DNCFILE <filename> before the "GO" command. Fonebuld can also have DNCFILE NONE and it will not check against the default Do Not Contact file.

You can now have up to four different DNC files for different reasons. In addition, you can control the status to use for numbers put in the DNC stack of the phone file. The parmfile syntax is:

```
DNCFILE: filename
DNCFILE: filename   ''second dncfile, etc.
```

In FONEBULD the syntax is:

```
DNCFILE = filename
```

or

```
DNCPREFIXFILE = filename
```

The default statuses are 96 and 70 respectively. In this way you can mark "do not contact" numbers coming from different sources with different statuses.

The DNC filenames used are stored in the phone header and are listed if you use a "HEADER xxxx" command in foneutil.

You can refer to it as the "DNCfile" in all commands. (This is the same as fone_number_index.)

You can use dncfile on the ~output command line in Mentor. For example:

```
~output mdnc.tr dncfile trfiledirectory=100000 append
```

This will make a DNC file for the CfMC phone system.

DNC FILE allows e-mail as well as phone numbers

You can now specify a file of e-mail addresses to check when building a sample file with FONEBULD to disallow records with those emails. The syntax is:

```
      Syntax:
      DNC_File: <filename>,e-mail
```

This is used in the parmfile and "DNCFILE=<filename>,email" in Fonebuld. In Fonebuld, use the command "Email_loc=<location.width>" to specify where the email address to compare is. Records that match will be given status 255 and saved in the "bad email" stack (340).

To build an e-mail Do Not Contact file, use the following Mentor syntax:

```
Ex:
      ~input <filename>,ASCII,Length=100
      ~output <e-mail dnc filename>,append emailindex writenow
      ~end
```

The maximum e-mail length is 128.

USING INTERNATIONAL PHONE NUMBERS IN THE DNC FILE

In order to accommodate international phone numbers, CfMC supports 18-digit phone numbers in the Do Not Contact file. CfMC creates a "fonenumber_index" which converts up to 18-digit numbers to an encoded 10-character case id. To build or append to a Do Not Contact file for numbers > 10 characters, use the following Mentor syntax:

```
      EX:
~input <ascii filename> ascii=100
~output <do not contact filename>trfiledirectory=<#####>
fonenumber_index maybecreate writenow
~end
```

This will read the first 18 characters of the file and convert them into a fonenumber_index. Even though the case ID is encoded, you can find the case in Mentor, using the command:

```
      next "123456789012345789"
```

The '12345, etc.' is the phone number you are looking for. Note that the phone number may contain special characters, such as - and (). The program handles these characters

REMOVING BLOCKS OF NUMBERS FROM THE AVAILABLE NUMBERS TO CALL

DO_NOT_CONTACT_PREFIX_FILE=xxxx is designed to remove cell phone numbers from the phone sample. The file is created exactly like the DO_NOT_CONTACT file, except that the program uses seven-digit case IDs. Any phone number that starts with those seven digits will be removed from the available sample when FONEBULD is run. The command may be invoked in FONEBULD for that run or in the parmfile as a system default. To build the file, add:

```
      EX:
~input <ascii filename> ascii=100 id=1.7
~output <dncprefix filename> trfiledirectory=<#####>
maybecreate writenow
~end
```

The <#####> is the number of phone number prefixes you expect to have. To invoke the file, in fonebuld say "DO_NOT_CONTACT_PREFIX_FILE= xxxx"where xxxx is the name of the TR file to use (the .tr extension is not required). To set a system default, add the command "DO_NOT_CONTACT_PREFIX_FILE: xxxx" in the CfMC parmfile. You can also shorten the command to "DNC_PREFIX_FILE".

Fonebuld gives numbers it resolves due to being in the "DNCPREFIXFILE" file a status of "70".

6.6.3 Preferred Time-of-Day Sampling

Some sample designs require specific scheduling of each call to the respondent, or different call scheduling for different groups of numbers. This can be accomplished using preferred time-of-day sampling. Preferred time-of-day sampling makes it possible to have each phone record be called in a different call-scheduling order. It uses the 'buckets' defined for each time zone, and is handled the same way NO ANSWER calls are handled, except that you decide which bucket the call will be returned to after each call, rather than simply defining a maximum number of times to call in each call period.

SPECIFYING THE TIME PERIOD CALL ORDER IN THE PHONE RECORD USING FONEBULD

To use preferred time-of-day sampling in FONEBULD, specify the preferred time-of-day location and length in the phone text area. You will need to put a string of digits between 1 and 9 in this field using Mentor or some other data processing software that tells the program in which bucket to place the number after the first call, the second call, and so, on for as many calls as you want to control. After the first call, the number will go into the bucket specified by the first digit. After the second call it will go into the bucket specified by the second digit and so on.

If you want to preschedule what bucket to start in, use columns 31-45 of the phone record with the text "BUCKET=#" in FONEBULD, where '#' is a number from 1-9 as above.

When you build the file or add the phone records specified using FONEBULD, the calling pattern will be established.

For example, in a designated field of the phone text you would provide a string, such as "123123778" to say after the first call, call back first in the morning, then the afternoon, then the evening/weekend, then the morning, then afternoon, then evening/weekend, and after 6 calls, call any time up to 2 more times, then go to All Targeted Attempts (ATA) bucket 8. This field can be different for each phone number, so one number can go into the MORNING bucket for the first four calls and AFTERNOON for the next four calls, while another number can just stay in MORNING,AFTERNOON,EVENING/WEEKEND bucket for all the calls made to it.

You can use preferred time-of-day control for some phone records, and let the system automatically control others. Once it is used on a particular phone record, however, it may not be turned off except by using the RELEASE_ALL_SYSTEM_CALLS parameter to release all numbers, or by Deleting the record using FONEUTIL, deleting the preferred time-of-day information out of the phone text, and adding the number back into the phone file using FONEBULD.

SURVENT ISSUES REGARDING PREFERRED TIME-OF-DAY SAMPLING

The buckets used by Preferred-Time-Of-Day sampling are:

Morning only	1
Afternoon only	2
Evening/weekend only	3
Morning or afternoon	4
Morning or eve/wkend	5
Afternoon or eve/wkend	6
Morning, Aft., Eve./Wkend	7
All targeted attempts	8
Bucket 9 storage	9

All Targeted Attempts means you hit the total number of attempts for each of the three target time periods, and you had allowed more total call attempts to possibly be made. Bucket 9 is another place to store numbers until they are released.

The three-day parts are called MORNING, AFTERNOON and EVENING/WEEKEND. They can be described more thoroughly as follows:

day part 1	day part time 1 through day part time 2 (Monday-Friday)
day part 2:	day part time 2 through day part time 3 (Monday-Friday)
day part 3:	day part time 3 through day part time 4 (Monday-Friday)
weekends:	day part time 5 through day part time 6 (Saturday) day part time 7 through day part time 8 (Sunday)

NOTE: The system will use day part times 5 and 6 for Sunday as well as Saturday if day part times 7 and 8 are both set to 12:00 am. The system will use day part times 1 and 4 for both Saturday and Sunday if day part times 5-8 are all set to 12:00 am. These definitions can be modified using Time Period Option 1-5 (see 2.2.1 The Phone Parameters).

Once Survent chooses a time zone to call from, it uses the buckets within a time zone to decide which number to call, along with the day part times (assigned in FONEBULD using the DAY_PART_TIME options), and the number of calls to day parts (TARGET1 through TARGET3).

When a call is attempted and gets a NO ANSWER status, Survent will by default assign it to be called back in the next open time slot based on the number of calls you have made to each time slot and the targets for each. But, if preferred time-of-day sampling is used, it assigns the number as you specified in your call scheduling string.

A number will go to the All Targeted Attempts (ATA) bucket in one of two situations:

1 All the targets are filled, so if you set targets at 2, 3, 2 and there have been two calls in the MORNING and EVENING and three calls in the AFTERNOON the number goes to ATA no matter what digits you may have in the preferred-time-of-day field (make sure you set these high enough for your preferred-time-of-day setting).

2 You put an “8” in the string of digits to specifically place the number in the ATA bucket. Using the previous example, make the phone text begin "4446668". This would be the best way to control exactly when the number should go into the All Targeted Attempts bucket.

6.6.4 Controlling the Sample with Markets

Often you will need to call sample that is controlled and reported on in groups, and you know the characteristics of each group in advance. We will call each of these groups a “market.” These groups are often geographic, such as north, south, east, and western regions, but can be based on anything we already know about the sample, type of business, income level, etc.

In order to use market controls, you must have a market name in each phone record. You could just assign a number to each market, and thus name them “001”, “002”, and so on, or you can give more meaningful names up to 8 characters long. Since the supervisors will be able to report on and control the sample by market, it is useful to have names that make sense. For instance, you may wish to control the sample in groups defined by sex of respondent by city (where the cities are New York, Atlanta, Boston, and Detroit). You would then create market areas such that the first character is M or F, followed by the city name NEWYORK, BOSTON, DETROIT, or ATLANTA. You could then have eight markets which would be MNEWYORK, FNEWYORK, MBOSTON, and so on.

Using market areas, you can control the relative amount of calling to a particular market during the study using market weights, either in FONEBULD initially, or in SURVSUPR while supervising, or programmatically in Survent. More particularly, this feature is usually used in conjunction with quotas such that when a market reaches its target quota, it will immediately shut down a market for calling. This has three distinct advantages over using quotas by themselves to control the sample:

- The numbers can be weighted up so that if you are falling short in a particular market's target sample, you can pick up more numbers without completely shutting down other markets using the HIDE command or setting their target quotas down.
- You can shut down a market using a market weight of 0. These numbers will then never be picked up by Survent, which will improve the study speed over using quotas to stop calling. Quota checks require you to first pick up the number in Survent, putting a heavy load on the system if you have met many of your quotas. Setting the market weight to 0 completely removes this load and the market is simply not called.
- · You will not have to use the HIDE and REVEAL options in SURVSUPR or FONEUTIL to stop calling a market, which takes time and CPU, is often diffic ult to specify and cannot easily be tracked.

Specific timed callbacks and numbers already in a special interviewer or ownership stack are not affected by any market weight other than 0. If the market weight is set to 0, and the system wants to call one of these numbers, it will automatically status the number with the call status that is assigned by the MARKETWEIGHT_ZERO_STATUS parameter in the server's PARMFILE (in the CONTROL directory or group under CFMC). If this value is not set, the program will send the number to an interviewing station, to let Survent determine what to do with this number.

For example, if you set MARKETWEIGHT_ZERO_STATUS: 213 in the PARMFILE the program will hide timed callbacks instead of dialing them in markets with a weight of zero.

You can get reports by market area in PHONERPT and see the bucket layout for each market in the supervisor program SURVSUPR.

SETTING UP THE MARKETS IN FONEBULD

Before building a market phone file, you must specify the set of markets to be used. The market in which a number is put will be specified in the phone text. Each record in a market will have the same string in the market location of the sample record.

To build the phone file with markets, run FONEBULD as always, but include the following commands in this order before the file is built.

```
MAXIMUM_MARKETS=<number>
```

If you want to allocate space for some maximum number of market areas that are going to be defined, use this. The number specified must be less than 960. If not specified, the number of MARKET commands (see below) for this phone file determines the maximum, and you will not be able to add any additional markets in the future without rebuilding the file.

The maximum number of markets depends on the number of time zones used. Markets times time zones times (10 buckets per time zone) must be <= 9650 and the maximum markets must be <= 960.

# time zones	maximum markets
1	960
2	480
3	320
2	240
5	192
6	160
7	137
8	120

```
TEXT_LENGTH=###
```

Tells the program where you can put the market location

```
MARKET_LOCATION=<column.width>
```

Says where in the phone text to get the market area specifier. The location must be between 51 and TEXT_LENGTH-50, that is, somewhere in the phone text area. The width is 1 if not specified, and may be 1-8.

```
MARKET = <name>,<number>,<weight>
```

This assigns the various markets. There should be one MARKET command for each market used in the study, or each string found in the MARKET_LOCATION in the phone records. If you say "MARKET=??? all phone records with no other market specified will be assigned to this market. If no "catchall" market such as this is specified, a number that does not have a matching market name will be an error, and be left out of the phone file.

<name> is the name of the market that is stored in the phone text. This may include any ASCII character except blank. The comparison ignores case of alphabetic characters.

<number> is the market area number. It can be left out in which case Fonebuld will assign the next available number. The number must be between 1 and MAXIMUM_MARKETS if specified.

<weight> is the initial calling weight, that is, how often a number will be chosen from this market relative to the other markets when a system call is made. If not specified it will default to 1. Weights can be a number from 0 to 9, just like the time zone weights. Generally weights will be 1 (to call the market) or 0 (to suppress calling to the market), but you may also slant the dialing to hit some markets more than others by changing the weight. Market weights can also be changed in FONEUTIL or SURVSUPR, or programmatically in the Survent interview by executing a PHONE,M,<market>,<weight> statement.

After the market commands are specified, enter “GO” to read the sample file in and have the phone file built.

If you are adding numbers to an existing phone file, you can also add new markets if space has been left for them by using the MAXIMUM_MARKETS command; specify the new markets exactly as before. If you say MARKET_LOCATION again, it must be the same as the first time. Be sure to specify a market number, as otherwise it will try to start at 1 again, which will already be in use. When you specify the number for a market, it must not be used by another market.

Here is an example command file to make markets with Fonebuld:

```
EX:
MARKETLOC=51.1
MAXMARKETS=10
MARKET A,1,1
MARKET B,2,1
MARKET C,3,1
MARKET D,4,2
GO
EXAM2
NUMBERS.RAW
QUIT
```

```
    QUIT
```

See example file MKTCV^SPX to convert an existing market controlled job into a non-market job or vice versa, or change the market numbers in an existing study.

REPORTING ON MARKETS AND CHANGING MARKET WEIGHTS

SURVSUPR and FONEUTIL have a MARKET command to show and/or change the weights for a particular market. The syntax for showing and/or changing weights is:

```
    MARKET <studycode> <market name> <-ZERO>
```

If the <market name> is specified you will see the timezone/bucket array for the specified market (and you can see others by using the + and - keys). If the <market name> is left off you will see a list of all the markets, the number of records in the market, and their weight, and you can modify market weights if you wish.

If "-ZERO" is specified, you will not see markets whose weights are currently set to 0.

Using Survent to Control Market Weights

Survent allows specific setting of the market weights. This is usually used when you check whether the target quota has been met for a particular market, and if so, to shut down the market by setting the weight to 0.

To control market weights, the PHONE,M statement is used. The syntax for controlling market weights is:

```
    {
    !IF condition
    !PHONE, M, <market name> ,<weight> }
```

You can specify a condition under which the weight will be set, and you specify the market name and weight on the question type line. Here is an example where you are setting the market weight to zero because the quota for that market is full:

```
      EX:
{MARKET:
!PHONE,G,51,1}
{
!IF MARKET$=”A” AND QUOTA(Market_A) >= QUOTA(Market_A.T)
!PHONE,M,A,0 }
```

This example says if the current interview is in market “A” and the quota for MARKET_1 is greater or equal to the target quota for MARKET_1, set the weight for that market to 0. See example files MARK^QPX and MARK^FBL which demonstrate how to set up and use markets in conjunction with quotas.

You can also use the !Phone,M statement to read the market name from the data.

```
      EX:
      !PHONE,M, [<label or location>], <weight>
```

The “label or location” would store the name of the market you will be updating. In this manner, you can update all your market weights with just the following two statements, instead of a statement for each market.

```
      EX:
{mkt: !phone,g,51,4}
{!if quotan([quotanum]) >= quotan([targnum])
!phone,m,[mkt],0}
```

This is in keeping with the increased number of markets we are supporting and allows you to work with quota names such that quota and market weight updates can occur with very few statements.

Controlling Nonsystem Numbers when a Market is Closed

Market weights only control system callbacks, so by default timed callbacks, special interviewer numbers, and owned numbers in a market continue to be released even if the market weight has been set to 0. But, if you want the server not to call these numbers in markets that are closed (because you set the market weight to 0 when quotas are full, for instance) you may do so using the MARKETWEIGHT_ZERO_STATUS parameter in the server’s PARMFILE. This tells the server that you want it to assign a status to these numbers instead of giving them to an interviewer or dialer to be called.

Using this keeps you from having to write code in your questionnaire to check whether you have a number in an over quota group to resolve after you have closed the market.

Likely statuses to use would be 188 or 213 to hide the numbers, 189 or 214 to place in bucket "9" (time zone holding area), or 11-69 to resolve the numbers. If the status is set to 0, or the command is not in the PARMFILE, the program will continue to hand the number to the interviewer or dialer. Statuses 83 (bad phone number), 186 and 211 (put in front of current stack), 187 or 212 (put in back of current stack) are not allowed.

NOTE: Like all PARMFILE commands, this is set for ALL studies under the server. You cannot assign a different status for these numbers from two different studies running under the same CfMC server at the same time.

6.6.5 Phone Says Datarecord Mode

“Phone Says Datarecord” mode means that you start with a partially filled in data record and add to it. This is especially useful for wave studies or studies with a “database” you want to continue updating.

To use this mode you put a case ID matching an existing data record starting in column 25 of the phone record. When Survent picks up a phone record, it looks to see if there is a case ID in that location, and if there is, it goes and gets the matching case from the data file.

It is very important that the data file be “indexed” if it has more than a few hundred records in it for faster access. Survent will build an indexed datafile by default, but in Phone Says Datarecord mode you must start with an existing file. If the file was not created by Survent, you can index the file by putting TRFILEDIRECTORY=# on the OUTPUT command of a Mentor run, where # is the number of cases you expect to have at most in the file. Here is a Mentor example to fix up an existing data file:

```
~input study^tr
~output new^tr trfiledirectory=20000 writenow
~end
```

Then, just rename new^tr to study^tr and put it in the interviewing area.

Be sure to build a data file with the CASE ID field filled. Even if the data is in the “case id” location, if the CASE ID field is blank, Survent will not find the case. Here is a Mentor example that will build the case id into the file starting from an ASCII file of information where the data is already formatted such that the case id is in columns 1-4:

```
~input CLIENT.DAT ascii=800 id=1.4
~output study^tr numcases=5000 writenow
~end
```

If you do not include the “id=1.4” statement, although there is a number in columns 1-4, Survent will not recognize that number as the case ID. Note also that the case ID must match EXACTLY what is in the sample file in columns 25+, including checking for leading zeroes.

If a case ID you specify in column 25 of the phone record does not exist in the data file, Survent will build a new data record to work with. So, you can have some sample with matching cases and other sample that is building new cases. If you do not want this to happen, add a statement in your questionnaire to check for existing data before moving forward in the interview, such as:

```
          {!if [1.4^^B]
          Case \[20.10] for phone number \[1.10] does not
          exist in the data file.
          Please tell your supervisor, terminating
          interview…
          !display}
          }!if [1.4^^B]
          !phone,s,78} ‘’give phone number resolved status
to

          keep from coming back
          {!if [1.4^^B]
          !spc,b} ‘’abort the interview
```

Also see the {!SPC,P} PREPARE command to get a specific case by case id or by using a conditional statement. This has the same mechanics, but is more often used for coding applications.

6.6.6 Ownership Mode

Ownership Mode allows you to assign certain phone numbers to particular interviewers. This typically occurs when there are long, involved interviews, or when there are particular interviewers that have the skills or knowledge needed to talk to a particular group of your sample.

Another use is to assign a number to an interviewer after they have started talking to the respondent, such as when the call is suspended. This allows the respondent to continue the interview with the same interviewer.

The idea is that a piece of sample can be owned by a specific interviewer, and if it is so owned then only the owner is ever going to see that number. We also provide facilities to override this ownership if necessary.

To summarize these features, sample can be initially assigned to an interviewer, or can be assigned once contact has been made. Once that sample is assigned, the system will only present it to the particular interviewer, unless the Ownership mode is turned off for the whole study or the specific number is requested by another interviewer.

If the specific number requested, the interviewer is warned that the phone record is owned by another interviewer. At that time, the interviewer may override the ownership for this call, in which case the call will still be owned by the other interviewer; or he may take the number for himself, in which case the number will belong to him from now on. Both of these features may be handled internally by the program or with special keywords by the interviewer.

NOTE: For webSurvent, the interviewer ID will be filled in as “WEBS” in call histories. This is so you can distinguish between histories saved from Survent, webSurvent or webCATI interviews.

INITIAL ASSIGNMENT OF THE PHONE RECORD TO A SPECIFIC INTERVIEWER

You initially assign the phone record to a specific interviewer using FONEBULD. There, place "ID:xxxx" anywhere between one space after the phone number and before column 22 of the initial phone record. xxxx would be replaced by the interviewer ID you want to assign to that record.

When the phone file is built with ID:xxxx in a phone record, the interviewer ID assigned is stored in the OWNER field in that record. If there is an ID in that field, when Survent searches for a number to call and that number is scheduled to be called, the interviewer with the ID specified will receive that number if they are available.

If that interviewer is not available, the number will go into the OWNED records stack until the interviewer signs on or the number is specifically released using FONEUTIL's option "B". The OWNED stack will be checked each time Survent goes to pick up a number to see if the interviewer asking for a number to call has a number available. If released, the number will be Rescheduled for the next day as if no call was ever attempted.

If you have many owned numbers in the phone file and few or none numbers that are not owned, you may cause the server to search far down when it seeks a number that can be given to the current interviewer. Ownership mode works best with either relatively small sample files (< 2000 numbers) or a low percentage of owned numbers in the file (20%).

USING SURVSUPR OR FONEUTIL TO CONTROL OWNERSHIP MODE

The phone file has a special switch called OWNER_MODE that can be set to YES or NO using FONEUTIL or SURVSUPR. If it is set to NO, the system behaves like it normally would if the phone records in the file were not assigned to a specific interviewer. This in effect releases all numbers that are still active to be able to be called by any interviewer on the system. If it is set back to YES, once again numbers assigned to specific interviewers will only be offered to those interviewers.

When in FONEUTIL and looking at a specific phone record with the F or # options, you can specifically change the owner, delete the owner (thus releasing the phone record to other interviewers), or just view the following information related to ownership mode:

OWNER: The current owner of the phone record, or, if blank, the record is available to any interviewer.

OLD_OWNER: If the record was owned by someone and the ownership was changed in any way, the last owner is recorded here.

OWNER_CHANGED: f the owner was changed in this record, this will indicate by whom. 1 means it was changed by an interviewer indicating "NEWOWNER" to get a record that was previously owned. 2 means it was changed by FONEUTIL.

OWNER_MODE: Is set to the value of OWNER_MODE In the phone file itself when the record was added to the phone file in FONEBULD. You may have some records in the file with OWNER_MODE set to YES and some set to NO, but this is just informational. The switch in the phone file header controls whether Ownership mode type operations are done to the file as a whole.

CONTROLLING THE OWNERSHIP OF PHONE NUMBERS IN SURVENT

As stated earlier, when getting phone numbers, only numbers which are not owned or which are owned by the interviewer requesting the number will be returned, unless a specific number is asked for. Once a phone record is available in Survent, new ownership may be established either programmatically or with the use of interviewer keywords.

In the case where a number is available, you can assign ownership to the current interviewer with the PHONE,8 question. Interviewers cannot assign a number to themselves by keyword; it must be set up by the Survent programmer to allow reassignment. Execution of the PHONE,8 writes the ID of the interviewer doing the interview into the owner field of the phone record.

A typical example of this is within the Suspend block, where you might want to assign ownership to this interviewer, since they have already established a rapport with the respondent. Here is an example of the specifications to do this:

```
EX:
{!SUSPEND}
{Setowner:
INTERVIEWER: Do you wish to retain this call as a
callback, or allow other interviewers to also make
the callback
!CAT
1 Retain call
2 Release to other interviewers }
{!IF Setowner(1)
!PHONE,8}
{!ENDSUSPEND}
```

In this example, the interviewer is asked whether they wish to retain contact with the respondent for future callbacks. If yes, the PHONE,8 execute and this interviewer is assigned as the new owner of the record.

Getting Numbers Owned by Others Using Standard Prompts

In the case where the number is specifically requested using the standard PHONE,5 question prompt, and the number is previously owned by someone else, a message will be returned that the number is owned by someone else, and the interviewer will be re-prompted for a phone number to call. At that point, they can enter one of two keywords to get access to the phone number:

OVERRIDE: The interviewer will get the number even though it is owned by someone else. However, the ownership of the number will not be changed in the case where the call is not completed and rescheduled.

NEWOWNER: This will retrieve the number and the ownership will be changed to the ID of the requesting interviewer. In this case, when the phone number is returned to the phone file as a complete or for rescheduling, the previous owner ID will be written to the OLD_OWNER field, and the OWNER_CHANGED field will be set to '1'.

Getting Numbers Owned by Others Using Internal Controls

The PHONE,5 statement has a format which records a code for the result of the attempt to find the number requested. You can then write code that will allow you to continue if you got a phone record, or reset and/or abort if you did not. (See 6.3.1 The PHONE Statement, PHONE,# Subtypes).

If you use this mode and the phone number is owned by another interviewer, the letters "OA" will be recorded in the phone status field, meaning 'owned by another'. Since there is no prompt at which the interviewer might enter override keywords, the OVERRIDE and NEWOWNER keywords can be programmatically executed by using the PHONE,6 and PHONE,7 questions respectively. Here is an example of controlling specific phone number access using internal controls:

```
      EX:
{Phonenum:

INTERVIEWER: Enter the phone number to get, or press
<Enter> to just get the next number.
!VAR,P,20,10}

{Phoncode: .2
!PHONE,5,,,Phonenum}
...
{Newowner:
!IF Phoncode$="OA"

INTERVIEWER:
The number is currently owned by another interviewer

(\[OWNER]).
Do you want to OVERRIDE the ownership at this time,
but let \[OWNER] have it later, or ...
Do you want to become the NEW OWNER of this phone record?
      !CAT
      1 OVERRIDE ownership for now
      2 Become the NEW OWNER
      3 Drop the record and get another }
{
!IF Newowner(1)
!PHONE,6}
{
!IF Newowner(2)
!PHONE,7}
...
```

In this example, PHONENUM is a VAR,P question which allows valid phone numbers including dashes (-) or parentheses around the numbers. The VAR,P will keep this as a 10-digit number, which is read by the PHONE,5 (because it is referenced as the place from which to get the phone number).

If the number returns a status code of 'OA', the interviewer is given the choice of overriding ownership, claiming ownership, or trying for another number. Notice that we can show the current owner from the phone record on the screen with \[OWNER].

If they decide to override or become the new owner, we execute the proper phone question to do so, and continue with the questionnaire.

6.6.7 Independent Validation of Interviews upon Completion

***NOTE*:** This validation feature may only be used with studies that use phone files in a servered environment. It works best when used with an SER/EIS predictive dialer. You may use this without the dialer, but the interviewer must transfer the call manually to the validator, which may cause management problems. This feature is not currently supported with a MSG dialer.

Your client or project designer may require that the information collected during the interview be validated by the supervisor or an alternate interviewer. This may be achieved at the end of the original interview by automatically transferring the call and the respondent’s information (data and phone record) to a designated person whom we refer to as a “validator.” There may be one or several validators per study. The validator will then execute a questionnaire that will only include the questions that are to be validated.

USING VALIDATION WITH A DIALER

CfMC software will work in tandem with the SER/EIS predictive dialer to validate interviews. The respondent's current call and their information will automatically be transferred to the next available validator.

Prepare Spec-writing Considerations

For the Interviewer's Questionnaire:

When a {!PHONE,X} statement is executed during the course of a questionnaire, the interview will be designated as one to be validated. It is recommended that this statement appear at the end of the interview to be sure that the interviewer cannot suspend or terminate the interview after the record has been flagged. It is also recommended that a !DISPLAY statement appear at the end of the interview telling the interviewer what will be happening next, for example:

```
…
{!phone,x}
{
Interviewer: Press Enter to end interview.
The call will be transferred to the next available
validator.
This may take a few moments.
!display
}
~end
```

For the Validator's Questionnaire:

The questionnaire for the validator will have the same study name in the questionnaire header as the original study since it will be using the same study files. Use the QFF_FILE_NAME= parameter to give the questionnaire file a different name. The validation questionnaire is designed such that the program will wait at the {!PHONE,4} statement for the next phone number to be sent it by an interviewer on the study. At that time the data record is also transferred and the questionnaire applied to the data collected by the interviewer. For example:

```
[MYSTUDY,CASE_LENGTH=800,COMMENT=” MYSTUDY &
Validation Qff”, QFFNAME=MYSTUDV]
{!PHONE,4}
{
Valname: 101.30 HIDE
!var,,30 }
{Valnum: 151.10
!var,,10}
{Valaddr: 201.40 HIDE
!var,,40 }
{
Hi, I would just like to make sure we have your
information correct.
Is this correct?

Name:             \|Valname
Telephone number: \|Valnum
Address:          \|Valaddr
!fld
1 YES
2 NO, REASK }
…
```

STARTING VALIDATOR INTERVIEWER SESSIONS

Validator interviewers not only must be started on the validation questionnaire but they must also be started with validation capabilities. These sessions may be initiated by a supervisor, or unsupervised interviewers may start themselves with such capabilities.

Sessions Initiated by a Supervisor (SURVSUPR):

In super (survsupr), start up designated validator interviewers as follows:

```
Enter a SUPERVISOR command--> atmv <interviewer list> <valqffname>
```

or

```
Enter a SUPERVISOR command--> atm <interviewer list> <valqffname> v
```

Once the validator interviewers start up, the supervisor will receive the normal startup line with "validator:1" included.

Unsupervised Interviewers:

Interviewers may put themselves in validation mode by entering "V" or "validator" mode after their interviewer IDs. They must also enter a "D" to designate that they are using the SER/EIS dialer. For example:

```
Type interviewer ID [,special type] or "quit" --> int101,d,v
```

On the EIS Gateway:

There will be a new EIS campaign dedicated to receiving calls for validation. The study name has a V postpended if the studycode is less than 8 characters, or the eighth character is replaced with a V. The phone call will automatically be transferred to the validator station at the end of the interview.

TRANSITION FROM THE ORIGINAL INTERVIEW TO THE VALIDATOR INTERVIEW:

The call will be transferred automatically as soon as the record has been sent to the validator. This may take a few moments. During this time, the call will remain with the original interviewer. The interviewer may be required to engage the respondent in idle conversation until the call is transferred.

The validator interviewer will execute the validator questionnaire up to the {!PHONE,4} statement. At the point where the {!PHONE,4} is executed, validators will see a progression of dots on their screens until they receive the record to be validated. Once the validation is completed, they will be required to start another interview and wait at the {!PHONE,4} once again (or the validation program could go there automatically).

USING VALIDATION WITHOUT A DIALER

Prepare Specwriting Considerations

For the Interviewer's Questionnaire:

When a {!PHONE,X} statement is executed during the course of a questionnaire, the interview will be designated as one to be validated. It is recommended that this statement appear at the end of the interview to be sure that the interviewer cannot suspend or terminate the interview after the record has been flagged.

It should also be explained to the interviewer that they should check to see if the validator is available. If the validator is available, the interviewer should end the interview and send the call to the validator's extension. The program will display a message indicating which terminal the phone and data record are being sent to This way, you know which extension to transfer to in the case there are multiple validators per study. Otherwise, allow the interview to finish as normal.

For example:

```
{
Interviewer: Check to see if validator is
available.
Select 1 if they are, 2 if they are not
!fld
1 Validator is available
2 Validator is not available
}
{
!if gotoval(1)
!phone,x}
~end
```

For the Validator's Questionnaire:

The questionnaire for the validator will have the same study name in the questionnaire header as the original study because it will be using the same study files. Use the QFF_FILE_NAME= parameter to give the questionnaire file a different name.

The validation questionnaire is designed such that the program will wait at the {!PHONE,4} statement for the next phone number to be sent to it by an interviewer on the study. At that time, the data record is also transferred, and the questionnaire applied to the data already collected by the interviewer.

For example:

```
[MYSTUDY,CASE_LENGTH=800,COMMENT=” MYSTUDY &
Validation Qff”, QFFNAME=MYSTUDV]
{!PHONE,4}
{
Valname: 101.30 HIDE
!var,,30 }
{Valnum: 151.10
!var,,10}
{Valaddr: 201.40 HIDE
!var,,40 }
{
Hi, I would just like to make sure we have your
information correct.
Is this correct?
Name:             \|Valname
Telephone number: \|Valnum
Address:          \|Valaddr
      !fld
      1 YES
      2 NO, REASK }
```

Starting Validator Interviewer Sessions

Validator interviewers not only must be started on the validation questionnaire but they also must be started with validation capabilities. These sessions may be initiated by the supervisor, or unsupervised interviewers may start themselves with such capabilities.

Sessions Initiated by Super (survsupr):

In super (survsupr), start up the designated validator interviewer as follows:

```
Enter a SUPERVISOR command--> sts <validator int id> <valqffname> v
```

Once the validator interviewer starts up, the supervisor will receive the normal startup line with “validtor:1” included.

Unsupervised Interviewers:

Interviewers may put themselves in validation mode by entering "V" or "validator" mode after their interviewer id. For example:

```
Type interviewer ID [,special type] or "quit" --> int101,v
```

Transition from Original Interview to Validator Interview:

The call must be manually transferred to the validator's phone extension by the interviewer at the end of the interview. This should be done only when the validator is available to receive the record.

The validator interviewer will execute the validator questionnaire up to the {!PHONE,4} statement. At the point where the {!PHONE,4} is executed, validators will see a progression of dots on their screens until they receive the record to be validated. When the validation is completed, the phone record is returned to the file with a default status code of 97 (you can change this to whatever status you want or to no status using the !PHONE,S statement). Once the validation is completed, they will be required to start another interview and wait at the {!PHONE,4} once again.

6.7 PHONE SYSTEM UTILITIES, REPORTS

There are several utility programs and batch files that can be run when using the phone system. FONEUTIL can change some phone file parameters, show you information about the phone file or specific phone records, and convert your phone file to an ASCII file to be added to other studies using FONEBULD. FONEUTIL can also repair a phone file that has gotten corrupted by a system crash or other non-standard termination.

SURVSUPR can modify phone file parameters, hide or reveal numbers, show phone information and set market weights while a study is live (see 4.4.2 *SURVSUPR Main Menu*).

PHONERPT generates 10 different summary reports, including interviewer productivity reports, current call status, and reports by market, special type, and replicate. MAKEZONE allows you to update the area code/exchange file to accept new area codes.

There are also many Mentor command files that work with the phone file. FONERUN, a command file in \CFMC\GO or CGO.CFMC, lets you automatically update, integrate, and do other options to all your phone files, files in a list, or a particular phone file. QUOTARPT reports number of completes, time on study and time to completion of quotas. Example files you may use include LGRPT^SPX to get detailed time on study information, FNCHK^SPX to make a thorough check of all fields in the phone file, or MKTCV^SPX to convert a MARKET phone file to a nonmarket phone file or vice versa.

6.7.1 Managing the Phone File with FONEUTIL

FONEUTIL can be used to change parameters (like SURVSUPR), to look at specific phone number records, to verify that the phone file is in good condition, or to convert the file to an ASCII file to be modified and read back into a phone file using FONEBULD, moved to another type of computer system, or used by other packages for processing.

To run FONEUTIL, enter:

```
FONEUTIL
```

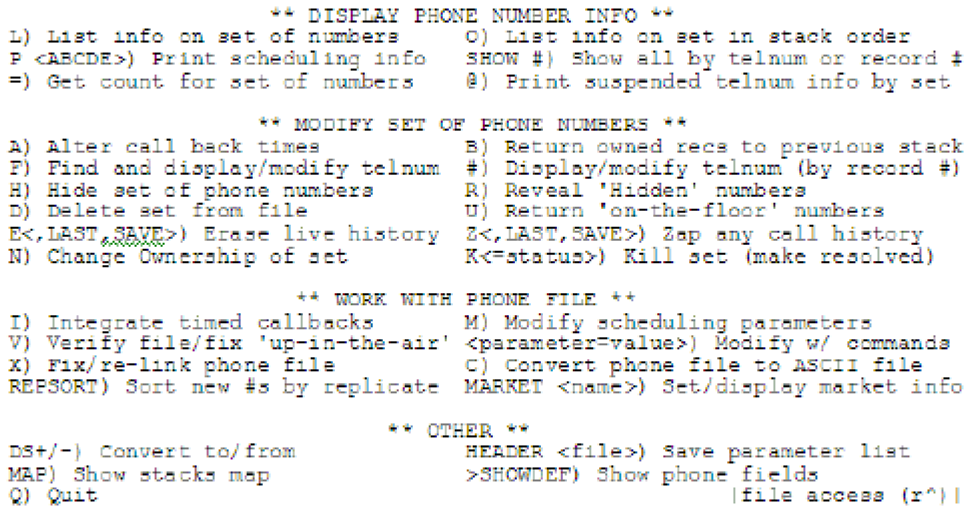
You will then be prompted for a list file, where you may press <ENTER> to work interactively, or enter a name if you want to save the output to a file. If you enter a name, much of the program output will only go to the file, and not to your screen. If you enter the following, the listed output will go both to the screen and to the file named FUTIL.

```
FONEUTIL CON "<filename>,ECHO"

EX:
FONEUTIL CON "UTIL,ECHO"
```

You will then be prompted for a study code. You may enter a study code or a fully qualified filename. If you enter a study code, the program will look to see if the CFMCFONE system variable is set. If it is, the program will look for the phone file in the directory here the CFMCFONE variable points. If it is not set, the program will look in your local directory for the file to open.

If you enter a full name, the program will look for the related index file and find it if it can (<study>^FNX); if it cannot, it will prompt you for the name. If you do not give a valid name, you will not be allowed to modify the file. If specified, you will have read write access to the phone file (unless it is currently open by the server or someone else). You cannot open a phone file in read-write mode if the study code in the phone file header doesn't match the file name. To open the file in read-write mode and make modifications, you must rename the file to the study code in the file or convert the file in FONEUTIL and read it back in using FONEBULD, providing the proper study code.

At this point, you may enter ***** at the prompt to see the menu of options:



```
                ** DISPLAY PHONE NUMBER INFO **
L) List info on set of numbers     O) List info on set in stack order
P <ABCDE>) Print scheduling info   SHOW #) Show all by telnum or record #
=) Get count for set of numbers    @) Print suspended telnum info by set

               ** MODIFY SET OF PHONE NUMBERS **
A) Alter call back times           B) Return owned recs to previous stack
F) Find and display/modify telnum  #) Display/modify telnum (by record #)
H) Hide set of phone numbers       R) Reveal 'Hidden' numbers
D) Delete set from file            U) Return 'on-the-floor' numbers
E<,LAST,SAVE>) Erase live history  Z<,LAST,SAVE>) Zap any call history
N) Change Ownership of set         K<=status>) Kill set (make resolved)

                  ** WORK WITH PHONE FILE **
I) Integrate timed callbacks       M) Modify scheduling parameters
V) Verify file/fix 'up-in-the-air' <parameter=value>) Modify w/ commands
X) Fix/re-link phone file          C) Convert phone file to ASCII file
REPSORT) Sort new #s by replicate  MARKET <name>) Set/display market info

                         ** OTHER **
DS+/-) Convert to/from             HEADER <file>) Save parameter list
MAP) Show stacks map               >SHOWDEF) Show phone fields
Q) Quit                                           |file access (r^)|
```

File access, in the lower right corner of the help screen, shows you what access you have to the current phone file. RW means Read-Write access, RO means Read Only.

In addition, FONEUTIL allows you to specify "READONLY" access so others can open the READWRITE file. By default, a .FON file is open in readwrite mode, so you that can make parameter modifications etc. To do so, when specifying the study or filename add ",readonly". For example:

Example:

```
<mystudy>,readonly
```

or

```
<mystudy>.fon,readonly
```

This is particularly useful if you are in the CFMC “fone” directory looking at things and there is a possibility that the study is about to be started by someone else, and you don't want to stop them.

You can also specify this in 'batch mode' applications for the same reason (you don't want to stop a study from starting).

NOTE: If the file is already open readwrite by some other process FONEUTIL will open the file readonly automatically.

SELECT STATEMENT

The “select” statement tells the program which records to include when processing the option specified. The following options will prompt for a SELECT statement:

Alter (A)

Change Ownership (N)

Convert (C)

Count (=)

Delete (D)

The maximum number of characters that can be processed is 32,000. This maximum would allow 2,500 10-digit phone numbers to be referenced.

The syntax for the SELECT statement is the same as the IF statement in PREPARE, see 2.6.2 *Using Data Location References* and 2.6.3 *Complex IF Statements* for a complete list of examples.

Here are three ways to say the same thing on a SELECT statement:

```
     Enter SELECT statement
     --> [Phone_number.3#415] AND [Special_type#1]
```

or

```
     --> [1.3#415] AND [22.1#1]
```

or

```
     --> @AREACODE=415 AND @SPECIAL_TYPE=1
```

Within brackets ([]s), you may use the label or location of the field of interest (see PHONE,G phone format description). You may refer to data as “string” type data by using [label$”string1”,”string2”,etc.] or “numeric” data using [label#range1,range2,etc.] where ranges may be simple numbers (1,23,35) or numeric ranges (1-23,45-99). You may use “AND” and “OR” to combine conditions. For long conditions, use an ampersand (&) at the end of the line to continue the specification on the next line.

Additionally, you may use previously defined keywords that are set up for you in FONEUTIL and SURVSUPR. Generally, for simple conditions using a whole field, you may wish to use the predefined variables (@VARIABLE_NAME), otherwise use the name or location in brackets ([]).

The keywords are read from \CFMC\CONTROL\PHONEDEF. This is an editable ASCII file that holds defined names and locations that you can use on the SELECT statement. You may add your own items to this file, using the existing items’ layout for your guide. Some of the items in the default PHONEDEF file look like this:

```
>DEFINE @AREACODE [1.3] ''Area code of phone number
>DEFINE @PREFIX [4.3] ''Prefix of phone number
>DEFINE @PHONE_NUMBER [1.10] ''10 digit phone number
```

To see the complete list of variables available, enter >SHOWDEF at a FONEUTIL prompt. A user can define any variable based on the record layout of the phone file (see *6.2.2 The Phone Record Format*).

Something defined in this file as >DEFINE AREACODE [1.3] can be referred to on the SELECT statement as:

```
@AREACODE=415 or
[AREACODE#415] or
[AREACODE]=415 or
[AREACODE $] = 415
```

When the operation is done, it will display how many numbers were read (total records in file), how many were selected (the number that matched your SELECT criteria), and how many were used (the number selected minus any unavailable numbers – those in use, resolved when action only applies to unresolved numbers, etc.).

Other Program Conventions

Options that scan through many numbers will print a message on the screen to give you an indication of their speed. Some options that will print several screens of information will not pause as the screen becomes full; you should designate a list file for the output (as shown on the previous page), or use **Ctrl-S** to stop the screen and **Ctrl-Q** to resume it.

Several options that look at many numbers give you the choice of using **Ctrl**-**Y** to get a status report while processing or to quit the operation. If you press **Ctrl**-**Y** while the option is working,

FONEUTIL will tell you what record it's currently on. This will give you an indication of how long the operation will take. Those options that support this feature will tell you at the beginning of their work. FONEUTIL will print a dot (.) for every 100 records processed.

Most commands have a shorthand and longhand version. Here are the commands and their two versions:

Shorthand	**Longhand**
*	Help
=	Count
@	list_suspend_stuff
%	show_stacks
#	record_number
>show_def	>show_defines
A	Alter
B	return_owned_records
C	Convert
D	Delete
DS+	spring_forward
DS-	fall_back
E	erase_history
F	find
G	generate_summary_info
H	hide
HEADER	ascii_phone_header
I	integrate_manana
K	kill
L	list
LOAD	load
M	modify_phone_header
MAP	map
MARKET	market
MARKETWEIGHT	marketweight
N	change_owner
O	display_stacks
P	print_rec_zero
Q	Quit

R	reveal
REPSORT	replicate_sort
S	gather_specials
SHOW	show_record
SORT_SPEC	sort_specials
SORT_SYS_CALLS	sort_system_calls
T	set_list_options
U	return_up_in_the_air
USE_FREE_STACK	use_free_stack
V	verify
X	fix
Z	zap_history

Here are the explanations of each command in help screen order:

DISPLAY PHONE NUMBER INFORMATION

L List set of phone numbers

The program will prompt you for a SELECT statement.

```
--> L
Enter SELECT statement or 'ALL'
--> [1.10]=4157775684
Ctrl-Y for status
415-777-5684 ACTIVE:FRESH ZONE:08 STATE:CA
(First call to this number...there is no history yet)
#Records read(135) #Selected(1) #Used(1)
Press any key to continue
```

Use the = (Count) option to just find out how many records match the SELECT statement.

O Display set in stack order

Enter the stack number or name. This option will print information for each phone number in this stack. Numbers are displayed in the order in which they will be called.

```
--> O
Enter STACK number, * for HELP, or QUIT
--> 350
(7) Record(s) in stack #350

513-888-0727 ACTIVE:FRESH ZONE:08 STATE:OH
513-888-9517 ACTIVE:FRESH ZONE:08 STATE:OH
513-888-4929 ACTIVE:FRESH ZONE:08 STATE:OH
513-888-8205 ACTIVE:FRESH ZONE:08 STATE:OH
513-888-3257 ACTIVE:FRESH ZONE:08 STATE:OH
513-888-7107 ACTIVE:FRESH ZONE:08 STATE:OH
513-888-9988 ACTIVE:FRESH ZONE:08 STATE:OH
(7) Record(s) in stack #350
```

In addition to a stack number, you can enter any of the following to display its stack:

COM/RES	completed/resolved numbers
DEL	deleted numbers
DUP	duplicate numbers
ERROR	“error” stack (bad phone records)
HID	numbers hidden in SURVSUPR or FONEUTIL or using statuses
MAN	“Mañana” numbers (timed callbacks, tomorrow or later)
Map	displays a map of all stack names and numbers and market counts
SPEC-#	special interviewer stack # (1-9)
TAPROX##	“approximate time” stacks (##=1-24, 1 for each hour of the day)
TIMED	displays all timed callback stacks that have at least one number
Thhmm	Specified timed stack, where time is hhmm
TZ##B#M###	System callback stacks for time zone # (0-24), bucket # (0-9) and market # (001-255). Leave the M### off if not using markets.

P Print phone file info screens

Displays up to six screens of phone file information. These are the same as the screens displayed in the SURVSUPR program using the SPI option. You may display any combination of the 6 screens using the letters “ABCDEF” to represent each screen. “P B” prints screen 2. “P ABC” prints screens 1, 2 and 3. “P BDE” prints screens 2, 4, and 5.

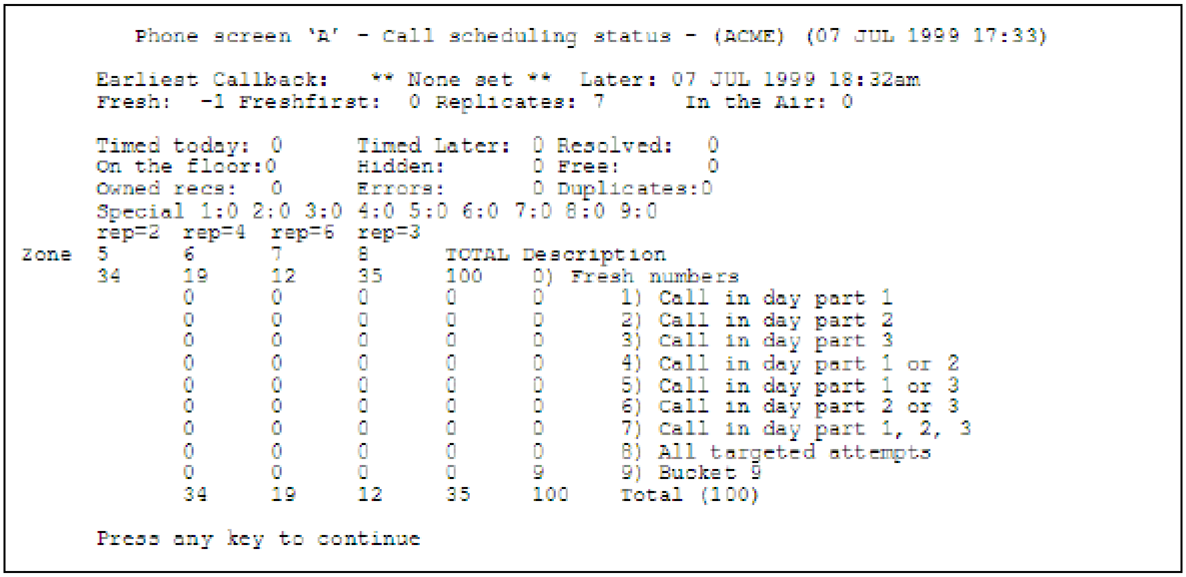
In addition, with the P option, you can phone information on a subset of markets or a particular market.

The new syntax is:

```
P <A-F> <study> <market list>
```

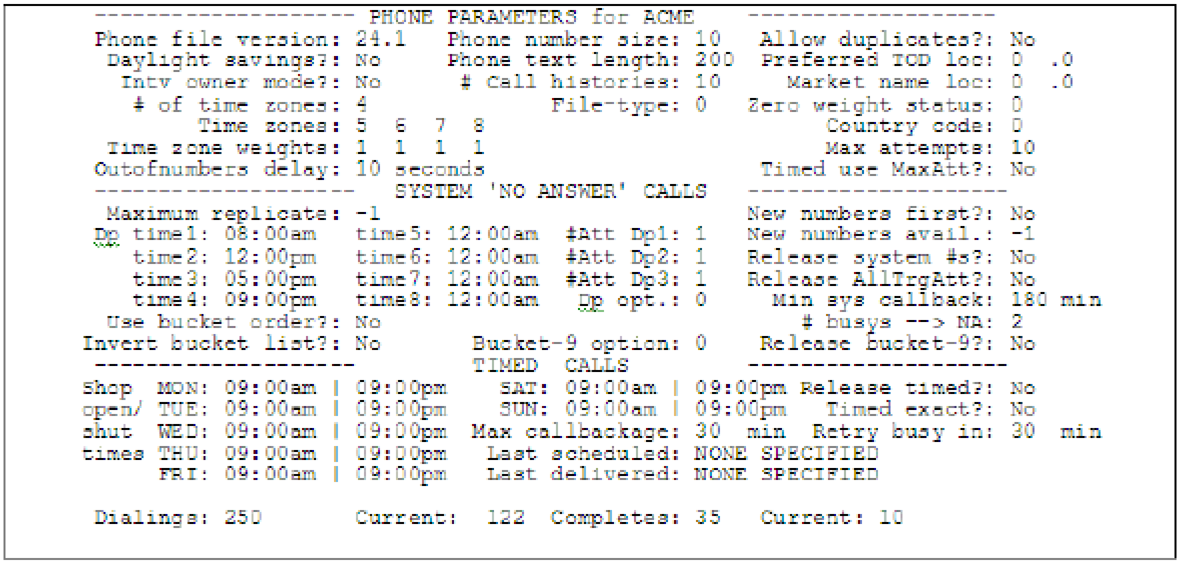
The "market list" can be any standard alpha reference using wildcards, such as "me*" for all markets beginning with me, or just a list of the markets you want to see, as in "north,south". This allows you to get summary numbers in the various screens - in particular the "F" screen (how many numbers available in stacks). Currently, the other screens don't get subset by market. Note that the same subsetting feature can be used in the Survsupr "SPI" option and the Fonesim "PZ" option (which are the same reports).

Screen 1: Summary information and count of numbers by stack and time zone



```
         Phone screen 'A' - Call scheduling status - (ACME) (07 JUL 1999 17:33)

      Earliest Callback:   ** None set **  Later: 07 JUL 1999 18:32am
      Fresh:  -1 Freshfirst:  0 Replicates: 7      In the Air: 0

      Timed today:  0     Timed Later:  0 Resolved:   0
      On the floor:0      Hidden:       0 Free:       0
      Owned recs:   0     Errors:       0 Duplicates:0
      Special 1:0 2:0 3:0 4:0 5:0 6:0 7:0 8:0 9:0
      rep=2  rep=4  rep=6  rep=3
Zone  5      6      7      8      TOTAL Description
      34     19     12     35     100   0) Fresh numbers
             0      0      0      0     0      1) Call in day part 1
             0      0      0      0     0      2) Call in day part 2
             0      0      0      0     0      3) Call in day part 3
             0      0      0      0     0      4) Call in day part 1 or 2
             0      0      0      0     0      5) Call in day part 1 or 3
             0      0      0      0     0      6) Call in day part 2 or 3
             0      0      0      0     0      7) Call in day part 1, 2, 3
             0      0      0      0     0      8) All targeted attempts
             0      0      0      0     9      9) Bucket 9
             34     19     12     35    100    Total (100)

      Press any key to continue
```

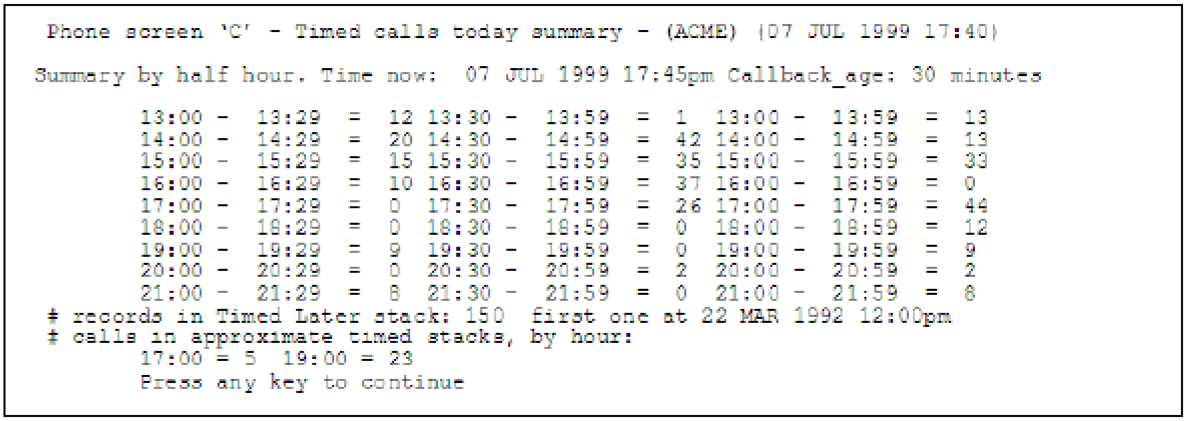
Screen 2: General information about parameter settings and counts



```
 -------------------- PHONE PARAMETERS for ACME    --------------------
 Phone file version: 24.1   Phone number size: 10   Allow duplicates?: No
  Daylight savings?: No     Phone text length: 200  Preferred TOD loc: 0  .0
   Intv owner mode?: No      # Call histories: 10     Market name loc: 0  .0
    # of time zones: 4              File-type: 0  Zero weight status: 0
         Time zones: 5  6  7  8                         Country code: 0
  Time zone weights: 1  1  1  1                         Max attempts: 10
 Outofnumbers delay: 10 seconds                     Timed use MaxAtt?: No
 --------------------  SYSTEM 'NO ANSWER' CALLS   --------------------
  Maximum replicate: -1                            New numbers first?: No
 Dp time1: 08:00am   time5: 12:00am  #Att Dp1: 1   New numbers avail.: -1
    time2: 12:00pm   time6: 12:00am  #Att Dp2: 1   Release system #s?: No
    time3: 05:00pm   time7: 12:00am  #Att Dp3: 1   Release AllTrgAtt?: No
    time4: 09:00pm   time8: 12:00am   Dp opt.: 0     Min sys callback: 180 min
  Use bucket order?: No                                 # busys --> NA: 2
Invert bucket list?: No       Bucket-9 option: 0    Release bucket-9?: No
 --------------------         TIMED  CALLS         --------------------
Shop  MON: 09:00am | 09:00pm    SAT: 09:00am | 09:00pm Release timed?: No
open/ TUE: 09:00am | 09:00pm    SUN: 09:00am | 09:00pm   Timed exact?: No
shut  WED: 09:00am | 09:00pm Max callbackage: 30  min  Retry busy in: 30  min
times THU: 09:00am | 09:00pm   Last scheduled: NONE SPECIFIED
      FRI: 09:00am | 09:00pm   Last delivered: NONE SPECIFIED

 Dialings: 250       Current:  122  Completes: 35   Current: 10
```

For a detailed explanation of hte information provided, see the PHONE PARAMETERS.

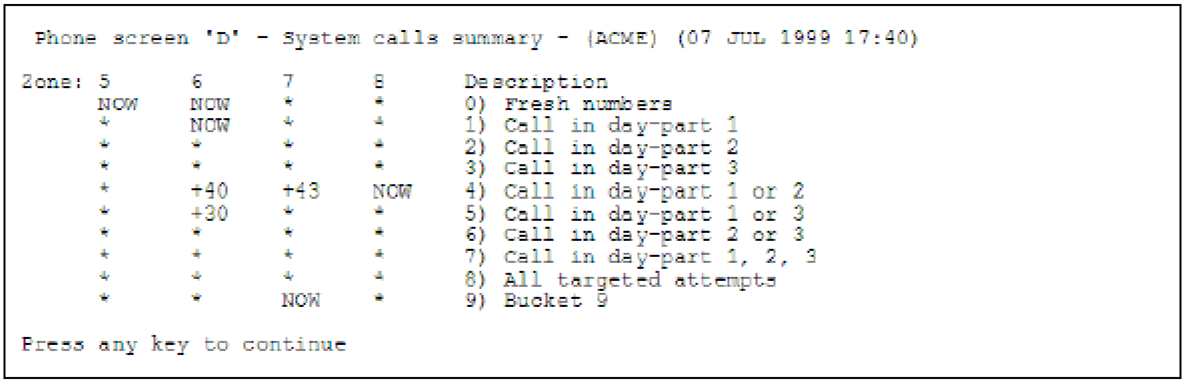
Screen 3: Summary of numbers in specific time callback stacks by half-hour



```
  Phone screen 'C' - Timed calls today summary - (ACME) (07 JUL 1999 17:40)

 Summary by half hour. Time now:  07 JUL 1999 17:45pm Callback_age: 30 minutes

        13:00 -  13:29  =  12 13:30 -  13:59  =  1  13:00 -  13:59  =  13
        14:00 -  14:29  =  20 14:30 -  14:59  =  42 14:00 -  14:59  =  13
        15:00 -  15:29  =  15 15:30 -  15:59  =  35 15:00 -  15:59  =  33
        16:00 -  16:29  =  10 16:30 -  16:59  =  37 16:00 -  16:59  =  0
        17:00 -  17:29  =  0  17:30 -  17:59  =  26 17:00 -  17:59  =  44
        18:00 -  18:29  =  0  18:30 -  18:59  =  0  18:00 -  18:59  =  12
        19:00 -  19:29  =  9  19:30 -  19:59  =  0  19:00 -  19:59  =  9
        20:00 -  20:29  =  0  20:30 -  20:59  =  2  20:00 -  20:59  =  2
        21:00 -  21:29  =  8  21:30 -  21:59  =  0  21:00 -  21:59  =  8
  # records in Timed Later stack: 150  first one at 22 MAR 1992 12:00pm
  # calls in approximate timed stacks, by hour:
        17:00 = 5  19:00 = 23
        Press any key to continue
```

This display shows the number of callbacks for today, broken down by half hour. “# records in Timed Later stack” gives the count of numbers scheduled for callback later than today. “# calls in approximate timed stacks, by hour” tells how many approximatetime calls there are by hour.

Screen 4: Display by stack of the number of minutes before the first number in each stack will be available with regard to minimum system callback time setting.

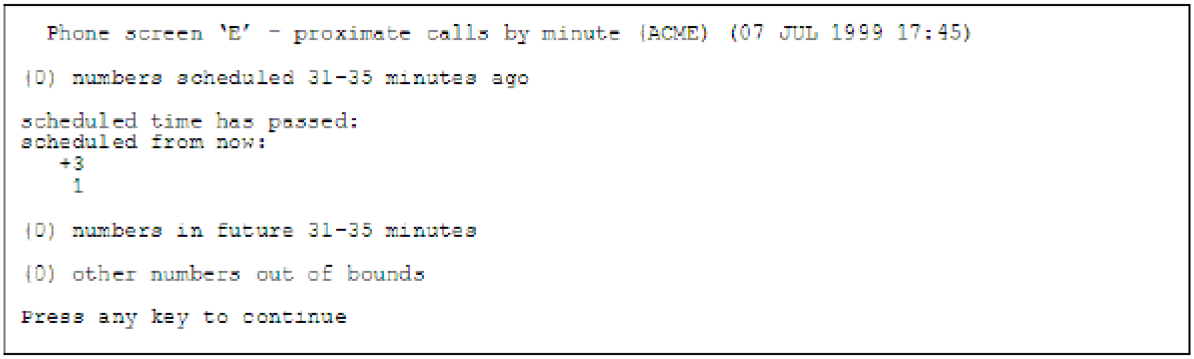


```
 Phone screen 'D' - System calls summary - (ACME) (07 JUL 1999 17:40)

Zone: 5       6       7       8       Description
      NOW     NOW     *       *       0) Fresh numbers
      *       NOW     *       *       1) Call in day-part 1
      *       *       *       *       2) Call in day-part 2
      *       *       *       *       3) Call in day-part 3
      *       +40     +43     NOW     4) Call in day-part 1 or 2
      *       +30     *       *       5) Call in day-part 1 or 3
      *       *       *       *       6) Call in day-part 2 or 3
      *       *       *       *       7) Call in day-part 1, 2, 3
      *       *       *       *       8) All targeted attempts
      *       *       NOW     *       9) Bucket 9

Press any key to continue
```

This displays the stack grid indicating times in +minutes from now for when the first phone number in each stack will be available. If the stack is currently available for callback, the word NOW appears instead of a number.

NOTE: The examples of screens 3 and 4 are not meant to reflect the same phone file as each other or as the other screen examples.

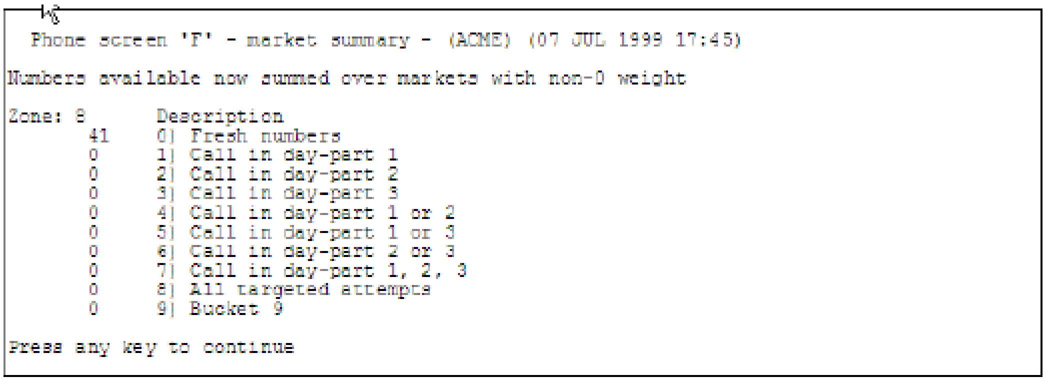
Screen 5: Show scheduled calls by minute



```
  Phone screen 'E' - proximate calls by minute (ACME) (07 JUL 1999 17:45)

(0) numbers scheduled 31-35 minutes ago

scheduled time has passed:
scheduled from now:
   +3
    1

(0) numbers in future 31-35 minutes

(0) other numbers out of bounds

Press any key to continue
```

This screen shows that there will be one call in three minutes from now and that no mbers have passed their call time.

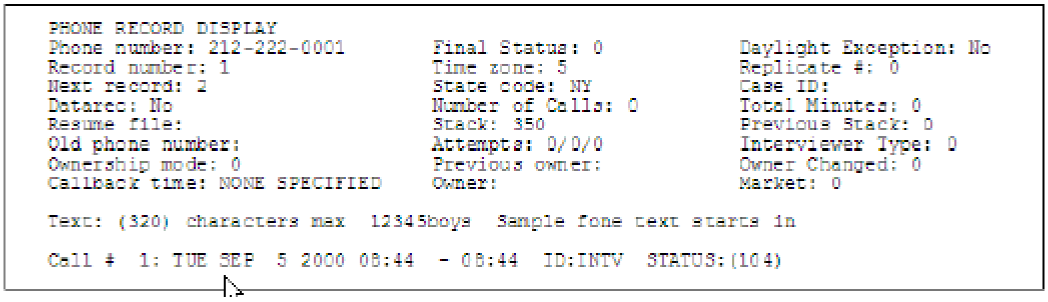
Screen 6: Display of the number of records in each stack across all non-zero weightmarkets



```
  Phone screen 'F' - market summary - (ACME) (07 JUL 1999 17:45)

Numbers available now summed over markets with non-0 weight

Zone: 8       Description
        41    0| Fresh numbers
        0     1| Call in day-part 1
        0     2| Call in day-part 2
        0     3| Call in day-part 3
        0     4| Call in day-part 1 or 2
        0     5| Call in day-part 1 or 3
        0     6| Call in day-part 2 or 3
        0     7| Call in day-part 1, 2, 3
        0     8| All targeted attempts
        0     9| Bucket 9

Press any key to continue
```

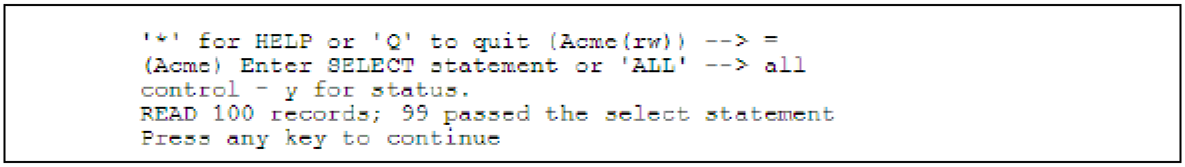
This displays the stack grid for the time zones used and tells you how many numbers are available across all markets. To see each market separately, see the MARKET command. See option M to modify phone file parameters.

SHOW # Shows the complete phone information for a particular number. You may say “SHOW <telephone number>” or “SHOW <record number>. “SHOW 415-777-2922” will show information on that phone number, and “SHOW 1” it will display record 1. The output will go to the list file if you have a list file open. This is the same as the “PHONE_NUMBER” command in the supervisor (SURVSUPR).



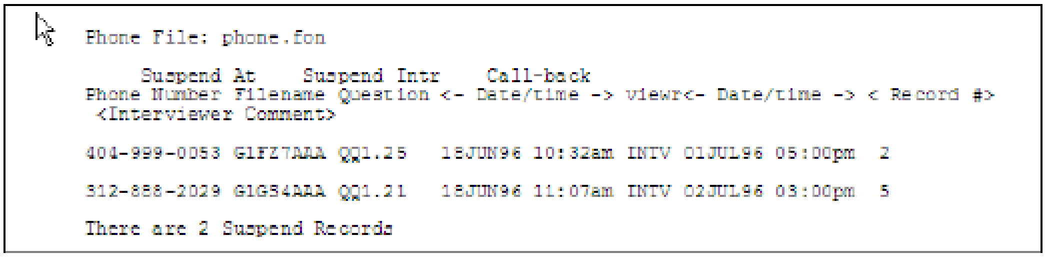
```
PHONE RECORD DISPLAY
Phone number: 212-222-0001        Final Status: 0             Daylight Exception: No
Record number: 1                  Time zone: 5                Replicate #: 0
Next record: 2                    State code: NY              Case ID:
Datarec: No                       Number of Calls: 0          Total Minutes: 0
Resume file:                      Stack: 350                  Previous Stack: 0
Old phone number:                 Attempts: 0/0/0             Interviewer Type: 0
Ownership mode: 0                 Previous owner:             Owner Changed: 0
Callback time: NONE SPECIFIED     Owner:                      Market: 0

Text: (320) characters max  12345boys  Sample fone text starts in

Call #  1: TUE SEP  5 2000 08:44  - 08:44  ID:INTV  STATUS:(104)
```

= Display a count of how many numbers meet your SELECT statement criteria'



```
'*' for HELP or 'Q' to quit (Acme(rw)) --> =
(Acme) Enter SELECT statement or 'ALL' --> all
control - y for status.
READ 100 records; 99 passed the select statement
Press any key to continue
```

@ Print information on suspended records

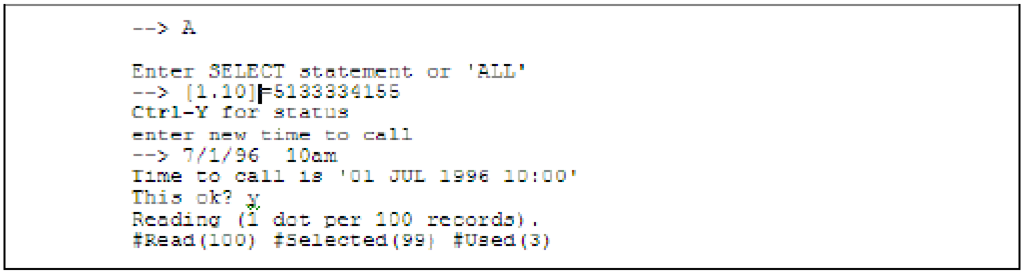


```
Phone File: phone.fon

     Suspend At     Suspend Intr    Call-back
Phone Number Filename Question <- Date/time -> viewr<- Date/time -> < Record #>
 <Interviewer Comment>

404-999-0053 G1FZ7AAA QQ1.25   18JUN96 10:32am INTV 01JUL96 05:00pm  2

312-888-2029 G1G34AAA QQ1.21   18JUN96 11:07am INTV 02JUL96 03:00pm  5

There are 2 Suspend Records
```

MODIFY SETS OF PHONE NUMBERS (FILES WITH READ-WRITE ACCESS ONLY)

A Alter callback time

This option allows you to change the time of numbers in the timed call stacks. You will be prompted with a SELECT statement. Then you will be presented with the first number to change and asked if the new time is ok. If you say "yes", it will continue to modify all the timed numbers selected.



```
--> A

Enter SELECT statement or 'ALL'
--> [1.10]=5133334155
Ctrl-Y for status
enter new time to call
--> 7/1/96  10am
Time to call is '01 JUL 1996 10:00'
This ok? y
Reading (1 dot per 100 records).
#Read(100) #Selected(99) #Used(3)
```

Enter a date and/or time in any of the formats described in 6.5 INTERVIEWING WITH THE PHONE SYSTEM.

FONEUTIL doesn't show you the numbers it is changing, so you might want to look at the callback stack first. Also, be careful when changing 'All' of the callback times. After changing callback times, you may want to re-Integrate the phone file.

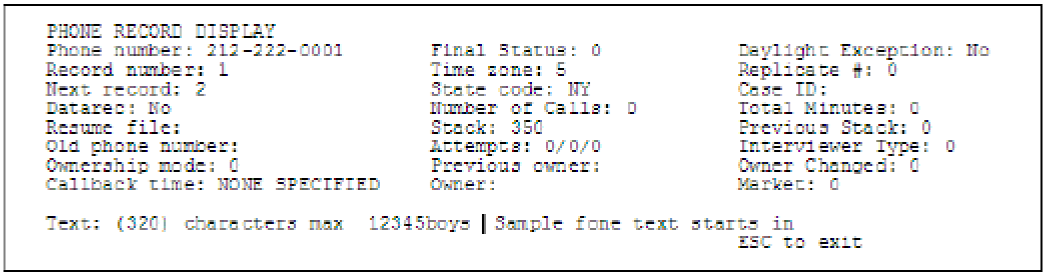
Alter will not work on active phone files.

B Return owned records to stack they came from

Puts numbers in the "owned records to call" stack (327) back into the stack from which they came. This is the temporary stack where numbers go if they are "owned" by a particular interviewer, come up to be called, but the interviewer is not on the system. ***NOTE:*** You cannot use the B option on an active phone file.

F Find, display, and modify one number

This option prompts you to enter a phone number. Enter it in the format of 4157770470 or 415-777-0470. If the phone number exists, FONEUTIL will display it along with its history (first one and last four call histories only):



```
PHONE RECORD DISPLAY
Phone number: 212-222-0001       Final Status: 0            Daylight Exception: No
Record number: 1                 Time zone: 5               Replicate #: 0
Next record: 2                   State code: NY             Case ID:
Datarec: No                      Number of Calls: 0         Total Minutes: 0
Resume file:                     Stack: 350                 Previous Stack: 0
Old phone number:                Attempts: 0/0/0            Interviewer Type: 0
Ownership mode: 0                Previous owner:            Owner Changed: 0
Callback time: NONE SPECIFIED    Owner:                     Market: 0

Text: (320) characters max  12345boys | Sample fone text starts in
                                                            ESC to exit
```

The “Call back time”, “Interviewer type”, and "Owner" can be altered using your arrow keys (on a monitor) or **Ctrl-U**, **D**, **R**, **L** (on a terminal) to position yourself, and then entering the new value.

When you press **ESC** to exit the screen, you will be asked if you want to see the rest of the phone text, or to confirm if you want to update the record. Answer T to see the phone text, Y to update, or N to leave the record unchanged.

The display of 320 columns of text starts at column 51 on the screen so that it matches up with the raw record layout. Also see the SHOW option that displays the information by record number and allows you to print it to a list file.

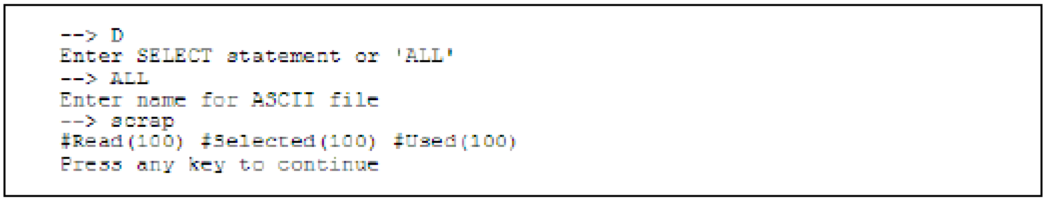
Find, display and modify one phone number by record

This works like option F, except you enter the absolute record number you wish to see from the phone file. The information for that record is then displayed. The “Call back time”, "Interviewer type”, and "Owner" can be altered like in F. You are also asked if you want to update the record or display all the text when you press ESC to exit.

Also see the SHOW option, which displays the information only by record number and allows you to print it to a list file.

D Delete set from file

Marks phone numbers as deleted and makes a copy of them in a backup file. You are prompted for a SELECT statement, and for the name of an output ASCII file for the deleted records. This can be a brand new file or an existing file to append to. These records can be modified and added back into the phone file with the FONEBULD program. See the PHONE,G statement for a description of the ASCII file format, and the FONEUTIL "C" option below for a discussion of exporting "squished" records to save disk space.



```
--> D
Enter SELECT statement or 'ALL'
--> ALL
Enter name for ASCII file
--> scrap
#Read(100) #Selected(100) #Used(100)
Press any key to continue
```

You cannot delete if the study is active. Note: Previously deleted numbers will be ignored by rs and FONEUTIL operations unless you use the "USE_FREE_STACK" parameter in FONEUTIL to temporarily make them available to be read. As you add new numbers though, they will use up "free" records such that you will not be able to recover those previously deleted records.

E Erase histories of live numbers

Used to reset unresolved numbers to "fresh" status, or remove a call history, without having to write the number out and re-read it in with FONEBULD. If you are doing call reporting, be sure to use the C option to save the call histories to a data file before erasing the histories in the phone file. You will see a warning before the call histories are erased.

You cannot erase histories if the study is active.

E,LAST will erase just the last history. Also see Z option to work on resolved numbers.

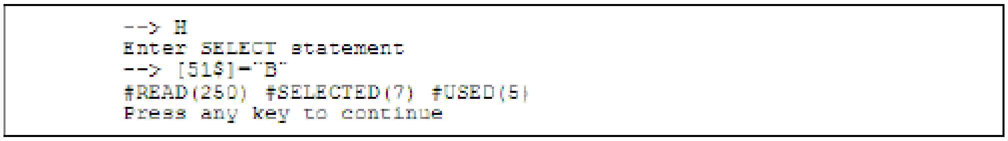
E,SAVE or E,LAST,SAVE will retain the histories even though the call is reset back to a fresh status or the last status of the number.

H Hide set of phone numbers

When you hide phone numbers, they are no longer available for calling. They are not removed from the phone file, but are put into stack 317. The program prompts for a SELECT statement and hides all of the numbers that meet that criteria. Hidden numbers can be called later if they are revealed (see option R).

Numbers in the following stacks cannot be hidden: already hidden, errors, duplicates, free records, resolved and any numbers which are up-in-the-air or active.

The following example illustrates how you could hide all business numbers currently being called.



```
--> H
Enter SELECT statement
--> [51$]="B"
#READ(250) #SELECTED(7) #USED(5)
Press any key to continue
```

You can later display the numbers that have been hidden by using option O and printing the numbers in the HID stack or stack 317. To hide numbers in an active study, use SURVSUPR’s HIDE option.

You cannot use “ALL" with HIDE because it usually makes no sense. If you want to hideALL the numbers, you can use a location specification, such as "[1#0-9]", which says all records with a number in the first position of the file which would be all numbers since that is where the phone number is.

NOTE: Hiding numbers can put a heavy load on the CfMC SERVER. If this affects your system performance, use market controls instead (see 6.6.3 *Controlling The Sample with Markets*). If you do not wish to use markets, see the example file BURN^QPX to have Survent hide numbers in a less intensive fashion.

K Kill set (force to be resolved)

This resolves the phone numbers selected and gives them a final status of 81 by default. Use "Kill=23" to kill a set of numbers and give them a resolved status of 23. This is useful to move a set of numbers to their "over quota" status or such. Numbers killed will not be called again unless you "Zap" the call histories. This option is preferable to the "D" option (delete numbers) in that the numbers will still be available for reporting. You cannot kill numbers if the study is active.

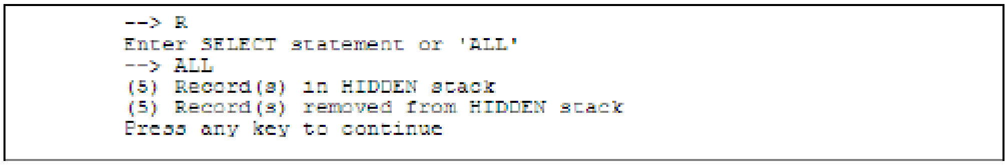
N Change ownership

Changes the owner to the ID specified for the selected group of records. You cannot use this option on an active study.

R Reveal hidden numbers

This reveals previously hidden (see option H) phone numbers. You are prompted for a SELECT statement. All numbers are revealed which meet the criteria of the SELECT statement. Revealed numbers are returned in order to the end of the stack they were in before being hidden. When revealing, you may wish to use the SORT_SYSTEM_CALLS option to make sure system callback stacks are sorted by the time of the last call to them.

If using replicates, you may need to use REPSORT to sort them in replicate order.



```
--> R
Enter SELECT statement or 'ALL'
--> ALL
(5) Record(s) in HIDDEN stack
(5) Record(s) removed from HIDDEN stack
Press any key to continue
```

To reveal numbers in an active study, use SURVSUPR's REVEAL option.

R,NOSORT will reveal numbers leaving replicates in their current order (even if that means they cannot be called due to bad order).

U Return on-the-floor numbers

Used to return numbers in the On-The-Floor stack (stack 328) into the stack that they were in before being Up-In-The-Air (in use) (See Verify option).

Z Zap histories (even resolved #s)

Used to reset numbers to "fresh" or their previous call stack, without having to write the number out and re-read it in with FONEBULD. Unlike option "E", this includes resolved numbers. If you are doing call reporting, be sure to use the C option to save the call histories to a data file before erasing the histories in the phone file. You will see a warning before the call histories are erased.

You cannot zap histories if the study is active.

Z,LAST will zap just the last history and return the call to its previous calling state. Use this, for instance, to release numbers that were resolved due to being over quota when you decide to get more interviews in that quota group.

Z,SAVE or Z,LAST,SAVE will retain the histories even though the call is reset back to a previous attempt.

Zap does not reset the numbers of calls, only the number of attempts. You cannot use "ALL" with Zap because it usually makes no sense. If you want to zap ALL the numbers, you can use a location specification, such as "[1#0-9]", which says all records with a number in the first position of the file, which would be all numbers since that is where the phone number is.

WORK WITH PHONE FILE

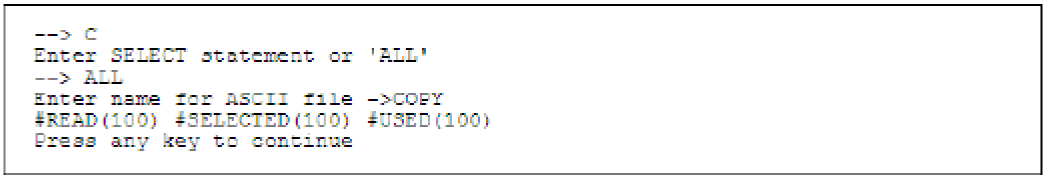
<parameter=value> Change phone parameter settings by keyword This option lets you modify the phone file header, like option M, but by keyword rather than on the screen. You can use the keywords as designated for FONEBULD, shown in 6.1.1, Building the Phone File. You can also do this by reading in a file created by the HEADER command in FONEBULD or FONEUTIL.

C Convert to ASCII file

This option writes out an ASCII file of records that match your SELECT statement. This can be a brand new file or you can append to an existing ASCII file. The file can then be read by other software programs or modified (i.e., using Mentor) and read back to this or some other phone file using the ASCII option. The original numbers are not deleted from the phone file; this makes a copy only.

The ASCII file that is created with the "C" option can be shortened by adding on the ASCII file name ,length=####, where #### is a number from 10 to the maximum length of the phone text of the phone file you are converting records from. This is good for users who want to get phone number lists or "raw" files created for use by other programs or to be loaded as new sample.

If you use the PARMFILE parameter ASCIIPHONERECTYPE: SQUISHED (in version 7.6l or later), the ASCII file will be built with a letter in column 50 representing the actual phone text length, and the file will be written such that the system information follows immediately after the phone text for 300 columns, after which the phone histories are written. Each letter increments the alleged text length by 200, For instance, a “B” in column 50 says that the phone text length is 400, “C” is 600, and so on. This “squished” file can then be read by FONEBULD as input. If you standardize your phone text length, you could use this for other programs as well, but you would have to define variables that matched the skewed locations.



```
--> C
Enter SELECT statement or 'ALL'
--> ALL
Enter name for ASCII file ->COPY
#READ(100) #SELECTED(100) #USED(100)
Press any key to continue
```

See *6.2.2 The Phone Record Format* for the layout of the ASCII file.

I Integrate timed callbacks

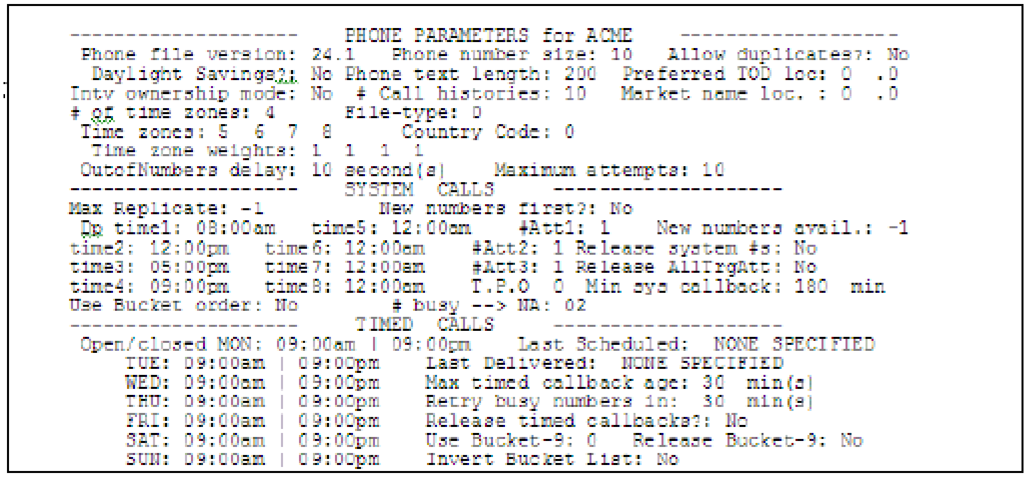
Sorts and integrates phone numbers in stack 313 (timed later), callbacks after today, into stacks 1-312, calls for today. Numbers that are not moved from stack 313 into 1-312 stay in stack 313. The program uses the current time and the CALLBACK_AGE to determine which numbers will be moved to the today stacks. Any number scheduled to be called at “current time – callbackage” will be moved.

If you do NOT integrate a phone file prior to the next day’s calling, when you log on to the study the next day the CfMC server will either not allow the study to start and force you to run FONEUTIL to integrate the file, or automatically integrate the file, depending on the setting of the “FORCE_INTEGRATE” parameter in the PARMFILE (in the CfMC CONTROL directory). If you want the server to always do integration for you, set FORCE_INTEGRATE to a very high number.

FORCE_INTEGRATE:0 server will never do automatic integration.

FORCE_INTEGRATE: 200 (default) do automatic integration for up to 200 numbers

M Modify scheduling parameters

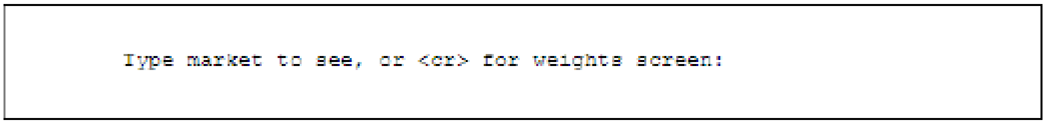


```
    --------------------    PHONE PARAMETERS for ACME    -------------------
     Phone file version: 24.1   Phone number size: 10   Allow duplicates?: No
      Daylight Savings?: No Phone text length: 200  Preferred TOD loc: 0  .0
    Intv ownership mode: No  # Call histories: 10   Market name loc. : 0  .0
    # of time zones: 4      File-type: 0
     Time zones: 5  6  7  8      Country Code: 0
      Time zone weights: 1  1  1  1
     OutofNumbers delay: 10 second(s)    Maximum attempts: 10
    --------------------    SYSTEM  CALLS    --------------------
    Max Replicate: -1          New numbers first?: No
     Dp time1: 08:00am   time5: 12:00am    #Att1: 1    New numbers avail.: -1
    time2: 12:00pm   time6: 12:00am    #Att2: 1 Release system #s: No
    time3: 05:00pm   time7: 12:00am    #Att3: 1 Release AllTrgAtt: No
    time4: 09:00pm   time8: 12:00am    T.P.O  0  Min sys callback: 180  min
    Use Bucket order: No        # busy --> NA: 02
    --------------------     TIMED  CALLS    --------------------
     Open/closed MON: 09:00am | 09:00pm    Last Scheduled:  NONE SPECIFIED
         TUE: 09:00am | 09:00pm    Last Delivered:  NONE SPECIFIED
         WED: 09:00am | 09:00pm    Max timed callback age: 30  min(s)
         THU: 09:00am | 09:00pm    Retry busy numbers in:  30  min(s)
         FRI: 09:00am | 09:00pm    Release timed callbacks?: No
         SAT: 09:00am | 09:00pm    Use Bucket-9: 0   Release Bucket-9: No
         SUN: 09:00am | 09:00pm    Invert Bucket List: No
```

See *6.2.1 The Phone Parameters* for information on these parameters.

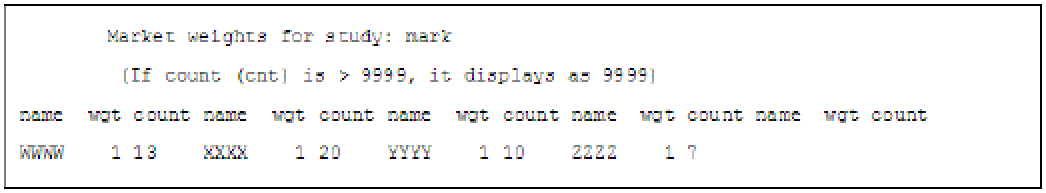
MARKET Display market info and/or modify market weights

This option lets you look at individual markets, look at totals by market, and/or change the market weights.



```
Type market to see, or <cr> for weights screen:
```

If you press <Enter>, you will get the market weights screen:

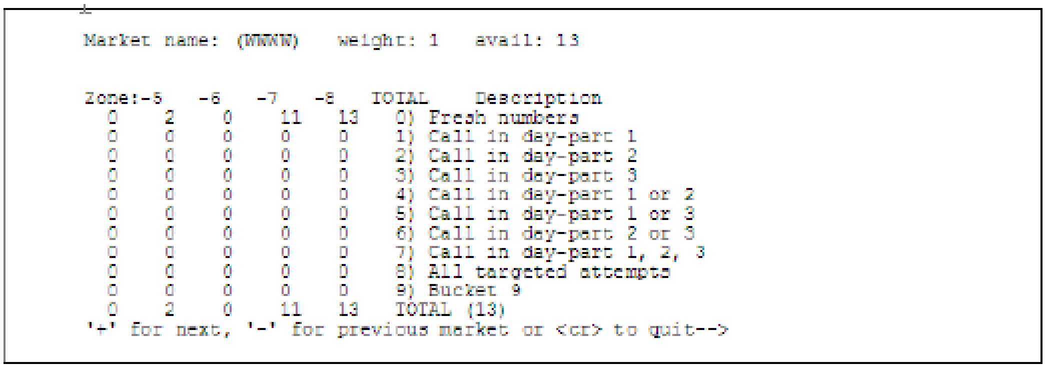


```
        Market weights for study: mark

         (If count (cnt) is > 9999, it displays as 9999)

name   wgt count name   wgt count name   wgt count name   wgt count name   wgt count

WWWW    1 13    XXXX    1 20    YYYY    1 10    ZZZZ    1 7
```

You may change each market's weight from 0-9 if the file is not active by using the arrow keys or **Ctrl-U**, **D**, **R**, **L** to move from field to field. Press **ESC** when done. Use the command "MARKETWEIGHT <marketname> = <0-9> to set this in batch mode.

If you have more than 100 markets, you will only be able to see the first 100 on the market weights screen. To see all market weight values, use the HEADER command to write out an ASCII record of all the weights. To see the number of records in each market, use the MARKET command and page through each market.

If you enter a market name, you will see its individual market screen:



```
Market name:  (WWWW)    weight: 1    avail: 13


Zone:-5    -6    -7    -8    TOTAL     Description
   0     2     0     11    13    0) Fresh numbers
   0     0     0     0     0     1) Call in day-part 1
   0     0     0     0     0     2) Call in day-part 2
   0     0     0     0     0     3) Call in day-part 3
   0     0     0     0     0     4) Call in day-part 1 or 2
   0     0     0     0     0     5) Call in day-part 1 or 3
   0     0     0     0     0     6) Call in day-part 2 or 3
   0     0     0     0     0     7) Call in day-part 1, 2, 3
   0     0     0     0     0     8) All targeted attempts
   0     0     0     0     0     9) Bucket 9
   0     2     0     11    13    TOTAL (13)
'+' for next, '-' for previous market or <cr> to quit-->
```

To be able to control up to 960 markets on one screen, more commands have been added to make it easier to move from screen to screen.

The commands are:

UNIX	DOS	Function
Ctrl-G	Home	Go to start of list
Ctrl-V	Page down	Go to next screen

Ctrl-T	Page up	Go to previous page
Ctrl-E	End	Go to end of list

Here is a market weight screen that displays 100 markets:

There is also a MARKETWEIGHT command (not shown on the help screen) that lets you change marketweights using command language, instead of only allowing changes on the interactive MARKET screen.

The syntax is:

```
MARKETWEIGHT <marketname>=<weight>
```

In addition, you can exclude the “ZERO WEIGHT” markets in SUPERVISOR and FONEUTIL by using the command:

```
MARKET <study> -ZERO (SUPERVISOR)
MARKET -ZERO (FONEUTIL)
```

EXPORTING MARKET WEIGHT VALUES TO ASCII

You can export market weight values to an ASCII file. By doing this, you can edit the file and read this back in using “&<filename>”. This allows you to have batch operations to change the weights or you can edit them in a file instead of on the market screen.

To do this, enter:

```
MARKET <study> MARKETWEIGHTOUT <filename>
```

(in the Supervisor)

or

```
MARKET MARKETWEIGHTOUT <filename>
```

(in Foneuti)

The file has lines saying MARKET <study> <marketname>= <weight>. If no filename is specified, it will write to the LPDEV file that is currently open.

REPSORT Sort numbers in replicate order

This option sorts the fresh number stacks in replicate order. This can come in handy if you added more new numbers to your phone file but think the replicate numbers may be out of order now. They must be in order for the program to release fresh number properly.

```
 '*' for HELP or 'Q' to quit (phone(rw)) --> repsort
Check rep order in stack 350, timezone 5
Check rep order in stack 360, timezone 6
Check rep order in stack 370, timezone 7
Check rep order in stack 380, timezone 8
 '*' for HELP or 'Q' to quit (phone(rw)) -->
```

V Verify integrity, return up-in-the-air numbers

Verifies the integrity of the phone file. It also recovers numbers which were left "up in the air" due to a non-standard termination of Survent while interviewing was in progress, and puts them in stack 328, where they are now considered "on the floor." See option U to put these numbers back into the regular stack that they came from.

```
--> V
Phone file is ok.  Done verifying phone file
```

ALWAYS verify a phone file after either a system crash or some other nonstandard termination of Survent. FONEUTIL quits if it finds more than 100 errors in the phone file during verification. To recover a file with more than 100 errors you must first use option X.

VERIFY also checks that the references to the resume files are correct, and that the resume files are intact. This may give you erroneous error messages if you have not set your CFMCRESUME variable before running FONEUTIL (DOS/UNIX).

A good practice is to run a daily batch job that verifies all phone files, produces a report, and subsequently adds back any "up-in-the-air” numbers. See\CFMC\GO\FONERUN.BAT (DOS) or /cfmc/go/fonerun (UNIX) to do this.

VERIFY CHECKDNC will go through numbers and move any it finds to the "Do not contact" stack (stack 330). This is to find numbers that either have been added recently to the "Do not contact" file or the phone number has changed. You can specify which type of numbers to check, by default it only checks new numbers. The options are:

Verify checkdnc fresh check new number stacks (default)

Verify checkdnc active check numbers that have not been resolved

Verify checkdnc all check ALL numbers

Verify,nosort will do a verify without sorting replicates (even if it means they can’t be called due to incorrect order).

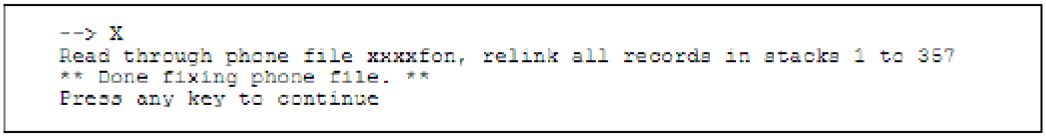
When you are done verifying, FONEUTIL can write lines to a file to say if there were any problems in the file. These lines can then be read and actions taken to resolve the problem.

To invoke this feature, say "setenv JCWFILENAME <filename>" and FONEUTIL will write the information there. Below is an example of the lines that might be written:

```
BADATDIALER       45    - 45  records still in the "atdialer" stack
BADSTACKS         1     - if any stack problems
BADREPLICATES     1     - if any stacks need repsort
BADUITACOUNT      10    - 10 records moved to onthefloor stack
BADRECORDCOUNT    5     - 5 records which failed checkrecord (BAD)
BADSUSPENDCOUNT   2     - 2 records with bad or missing suspend file
BADERRORCOUNT     6     - 6 records in error stack at end of verify
```

See the Mentor command file FNCHK.SPX as well; that will fix up phone records if it finds problems and save the bad records into a “BADRECS” file for you to fix or otherwise decide what to do with them. You would take the output file from FNCHK.SPX called GOODRECS and build a new study .fon file from it and it will be error-free.

X Fix, re-sort, and re-link the phone file.



```
--> X
Read through phone file xxxxfon, relink all records in stacks 1 to 357
** Done fixing phone file. **
Press any key to continue
```

This fixes records with invalid system information in them and resets the pointers from record to record in a stack. It is generally only needed if there is an abnormal termination of the study, but using the FONERUN batch command file as a nightly job to Fix and Verify your phone files will guarantee that your studies are ready to run each morning.

If some records cannot be fixed, run Mentor using the example file FNCHK^SPX to tell you which fields are incorrect and try to fix them for you. Read the “fixed” ASCII file back into FONEBULD

to build a new file using the ASCII option.

NOTE: FIX automatically resorts records in the “new” buckets by replicate if you are using replicates (like REPSORT) and numbers in the system callback buckets by time of call (like SORT_SYSTEM_CALLS).

Other Commands:

>SHOWDEF Show SELECT statement >DEFINEd items

This displays the phone file format defined. See SELECT STATEMENT earlier in this section for details on this option.

DS+ Convert to Daylight Savings Time

Changes exception numbers from “daylight savings” time to “standard” time. This should only be done once a year, in the fall, on active jobs. It is assumed that you will have adjusted the system clock back 1 hour, so this will only change the numbers that don’t observe daylight saving time.

DS- Convert to Standard Time

Changes exception numbers from “daylight savings” time to “standard” time. This should only be done once a year, in the fall, on active jobs. It is assumed that you will have adjusted the system clock back 1 hour, so this will only change the numbers that don’t observe daylight saving time.

HEADER Write out scheduling parameters

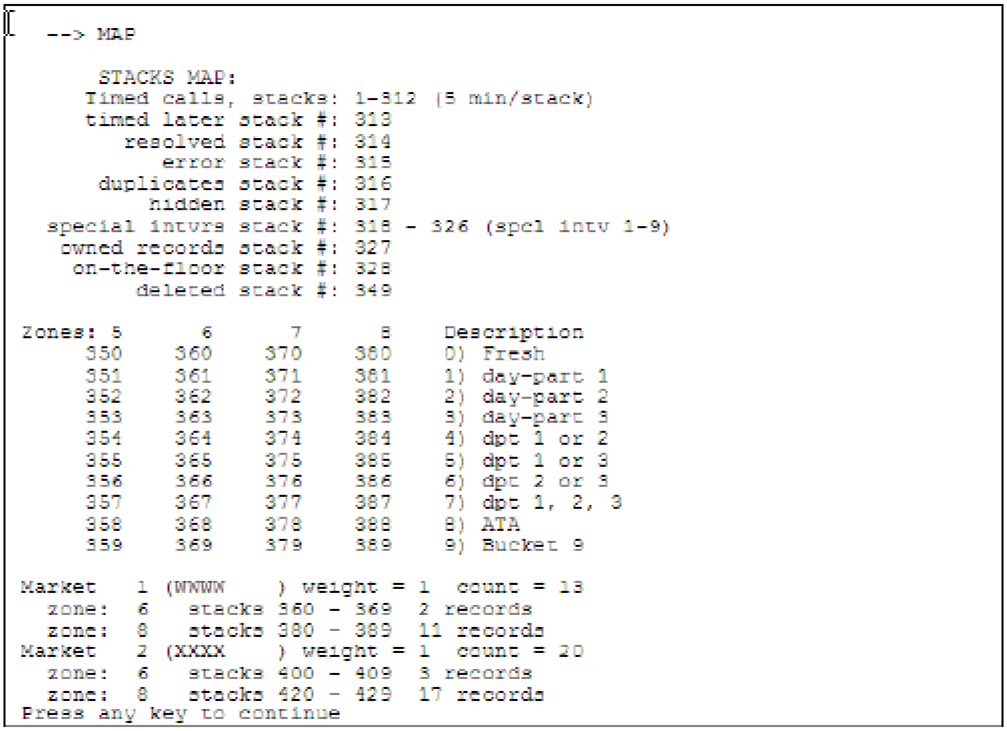
This writes an ASCII version of the phone scheduling parameters to a file, so you can read it or use it in batch operations. The commands can be read into FONEBULD or FONEUTIL. This command is also in the FONEBULD program. See the example file PHONE^CMD for typical header output.

HOLD_SPECIALS: YES causes FONEUTIL to keep numbers in special stacks when integrating the file.

This tells the system not to perform FONEUTIL;s “integrate” function, which takes timed numbers in the 9 special stacks that are too old to call and it moves them back into the "call later" stack. In other words, just leave them in the "special" stack until they are called by a special interviewer.

MAP Show a map of the stacks

```
    --> MAP
```



```
  --> MAP

      STACKS MAP:
     Timed calls, stacks: 1-312 (5 min/stack)
     timed later stack #: 313
        resolved stack #: 314
           error stack #: 315
      duplicates stack #: 316
          hidden stack #: 317
  special intvrs stack #: 318 - 326 (spcl intv 1-9)
   owned records stack #: 327
    on-the-floor stack #: 328
         deleted stack #: 349

Zones: 5      6      7      8     Description
     350    360    370    380     0) Fresh
     351    361    371    381     1) day-part 1
     352    362    372    382     2) day-part 2
     353    363    373    383     3) day-part 3
     354    364    374    384     4) dpt 1 or 2
     355    365    375    385     5) dpt 1 or 3
     356    366    376    386     6) dpt 2 or 3
     357    367    377    387     7) dpt 1, 2, 3
     358    368    378    388     8) ATA
     359    369    379    389     9) Bucket 9

Market   1 (WWWW     ) weight = 1  count = 13
  zone:  6   stacks 360 - 369  2 records
  zone:  8   stacks 380 - 389  11 records
Market   2 (XXXX     ) weight = 1  count = 20
  zone:  6   stacks 400 - 409  3 records
  zone:  8   stacks 420 - 429  17 records
Press any key to continue
```

If you have markets, the map will include all market stacks.

PHONE_MOVE_RECORDS Allows you to move records from stack to stack using a "SELECT" statement to determine which records to move. Used in conjunction with the "Call_first" stack, this lets you give numbers immediate priority as well, if you choose. The syntax is:

```
                Syntax:
PhnMv Fromstack=<from stacks> Tostack=<to stack> <select statement>
```

Stacks can be numbers or names. You can use up to 10 numbered stacks in the “from” field, but only one named stack.

Here are possible names:

From Names	To Names	Description
#	# (<350)	Stack number
CALLFIRST	CALLFIRST	“Call first” stack
FRESH	FRESH	
EMAIL_INVITE/REMIND	EMAIL_INVITE/REMIND	
TZ#B#		Time zone X, Bucket Y
TZ#B#M[#/marketname]		Time zone X, Bucket Y, market Z

Where "#" is a stack number. Stack numbers go from 1 - 9650. Notice that you can move things FROM most stacks, but you cannot move things to stacks > 350 (the time zone/market array). Users shouldn't move things from time zone to time zone or market to market in this way.

The "TZ#B#" syntax means "Time Zone X, bucket Y". The TZ#B#M# means "Time Zone X, bucket Y, Market Z". You can substitute a market name for the market number.

Also, there is a keyword "older_than" that will allow you to say "older_than 3 days" (to only get older records). Here are some examples:

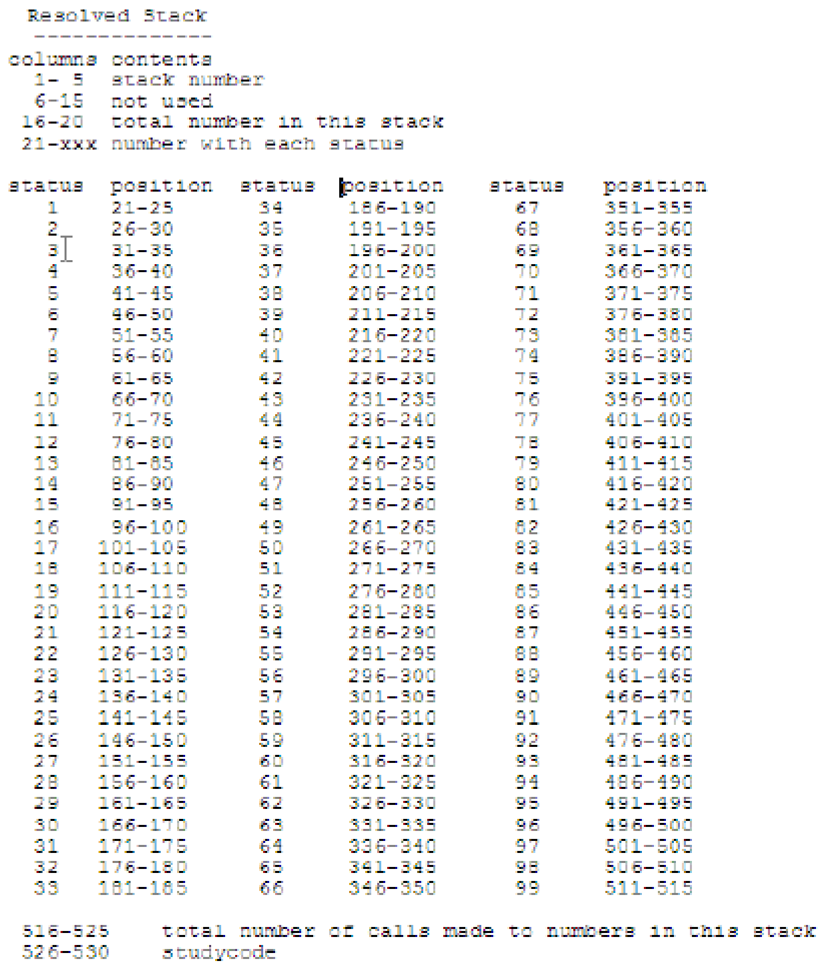
```
phone_move_records fromstack=fresh tostack=callfirst [52.3$]="001"
phone_move_records fromstack=callfirst tostack=emailsend 1=1
phone_move_records fromstack=email_invite tostack=email_remind 1=1
phone_move_records fromstack=tz5b3mMymarket tostack=fresh 1=1
phone_move_records fromstack=tz5b3 tostack=fresh older_than 3 days 1=1
phone_move_records fromstack=fresh tostack=callfirst[1.10$]="4069954724"
phone_move_records fromstack=350 360 tostack=callfirst 1=1
```

Q Quit working with current phone file.

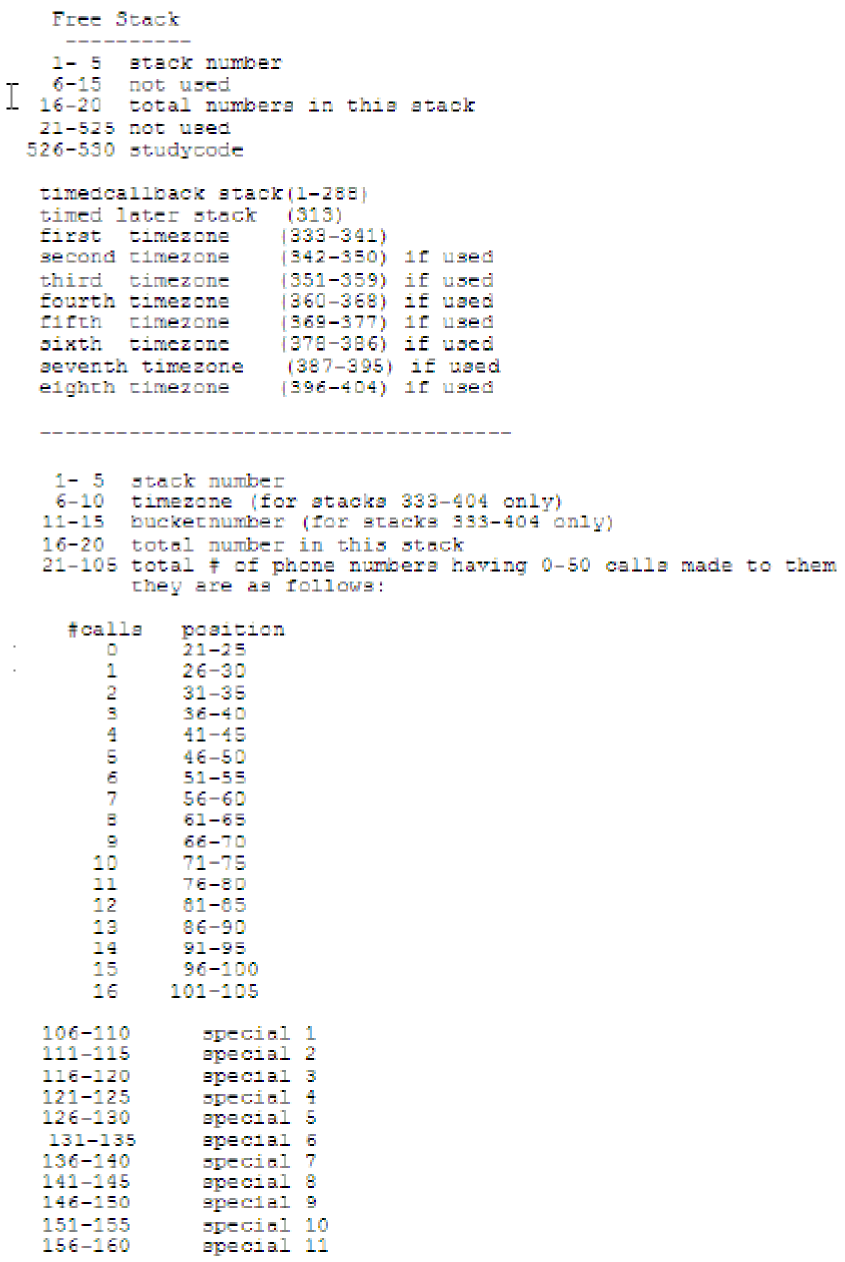
Additional Commands Not on the Help Screen

G This generates summary records.

It creates an ASCII file named SUMMARY, which tells you how many records are in each stack and how many have each status. The format is:



```
  Resolved Stack
  --------------
columns  contents
  1- 5   stack number
  6-15   not used
 16-20   total number in this stack
 21-xxx  number with each status

status   position   status   position     status    position
    1     21-25        34     186-190        67      351-355
    2     26-30        35     191-195        68      356-360
    3     31-35        36     196-200        69      361-365
    4     36-40        37     201-205        70      366-370
    5     41-45        38     206-210        71      371-375
    6     46-50        39     211-215        72      376-380
    7     51-55        40     216-220        73      381-385
    8     56-60        41     221-225        74      386-390
    9     61-65        42     226-230        75      391-395
   10     66-70        43     231-235        76      396-400
   11     71-75        44     236-240        77      401-405
   12     76-80        45     241-245        78      406-410
   13     81-85        46     246-250        79      411-415
   14     86-90        47     251-255        80      416-420
   15     91-95        48     256-260        81      421-425
   16     96-100       49     261-265        82      426-430
   17    101-105       50     266-270        83      431-435
   18    106-110       51     271-275        84      436-440
   19    111-115       52     276-280        85      441-445
   20    116-120       53     281-285        86      446-450
   21    121-125       54     286-290        87      451-455
   22    126-130       55     291-295        88      456-460
   23    131-135       56     296-300        89      461-465
   24    136-140       57     301-305        90      466-470
   25    141-145       58     306-310        91      471-475
   26    146-150       59     311-315        92      476-480
   27    151-155       60     316-320        93      481-485
   28    156-160       61     321-325        94      486-490
   29    161-165       62     326-330        95      491-495
   30    166-170       63     331-335        96      496-500
   31    171-175       64     336-340        97      501-505
   32    176-180       65     341-345        98      506-510
   33    181-185       66     346-350        99      511-515

 516-525    total number of calls made to numbers in this stack
 526-530    studycode
```



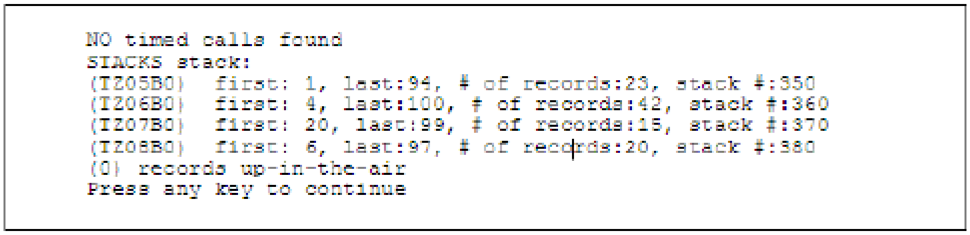
```
   Free Stack
    ----------
   1- 5  stack number
   6-15  not used
  16-20  total numbers in this stack
  21-525 not used
 526-530 studycode

  timedcallback stack(1-288)
  timed later stack  (313)
  first  timezone    (333-341)
  second timezone    (342-350) if used
  third  timezone    (351-359) if used
  fourth timezone    (360-368) if used
  fifth  timezone    (369-377) if used
  sixth  timezone    (378-386) if used
  seventh timezone    (387-395) if used
  eighth timezone    (396-404) if used

  --------------------------------------


   1- 5  stack number
   6-10  timezone (for stacks 333-404 only)
  11-15  bucketnumber (for stacks 333-404 only)
  16-20  total number in this stack
  21-105 total # of phone numbers having 0-50 calls made to them
         they are as follows:

    #calls   position
         0     21-25
         1     26-30
         2     31-35
         3     36-40
         4     41-45
         5     46-50
         6     51-55
         7     56-60
         8     61-65
         9     66-70
        10     71-75
        11     76-80
        12     81-85
        13     86-90
        14     91-95
        15     96-100
        16    101-105

  106-110      special 1
  111-115      special 2
  116-120      special 3
  121-125      special 4
  126-130      special 5
   131-135     special 6
  136-140      special 7
  141-145      special 8
  146-150      special 9
  151-155      special 10
  156-160      special 11
```

MARKET ALLMARKETS This prints the standard time zone grid (like a phone screen A) for all markets to the listfile and or screen.

MARKET PRINTWEIGHTS This prints one line per market, listing its market number, name, current weight and number of people in the market's available call grid.

MARKET_WEIGHT The command "Market_weight <market>=<0-9> will set the weight the same way as you can set it interactively on the market weights screen (see MARKET above). This is provided so you can update markets in batch applications.

% Shows summary stack counts, as follows:



```
NO timed calls found
SIACKS stack:
(TZ05B0)  first: 1, last:94, # of records:23, stack #:350
(TZ06B0)  first: 4, last:100, # of records:42, stack #:360
(TZ07B0)  first: 20, last:99, # of records:15, stack #:370
(TZ08B0)  first: 6, last:97, # of records:20, stack #:380
(0) records up-in-the-air
Press any key to continue
```

This shows the first and last record number in the stack and # of records for each stack with records in it.

S This command looks for numbers that pass the "SELECT" statement specified in the scheduled stacks that are marked for special interviewers, and "gathers" them to the special interviewer stacks. You may wish to do this for reporting purposes (so you can see all the scheduled numbers) or to immediately make those numbers available to your "special "interviewers.

SORT_SPECIALS This command sorts the special interviewer stacks by time zone; the idea is to call certain time zones before others. You may say "SORT_SPECIALS 1-3,5" to only sort the special stacks for interviewer types 1-3 and 5 (there are 9 possible “special interviewer types”, so this parameter can refer to types 1-9).

SORT_SYSTEM_CALLS This sorts the system callback stacks by the time of the last call to that number, so that numbers will come up for calling in the proper order. This is particularly important if you use the “X” option, which sorts the number is ‘record number’ order. If you use the “X” option in the middle of a shift and do not use this parameter, stacks may be unusable for a while because the ‘minimum system callback time’ can prevent calls that were recently called and sorted to the front of their stack from coming up.

T Tell what to print in L and O (Set_list_options)

This displays a list of display options. These display options will affect FONEUTIL options L and O. Choose the ones you want for the current run of FONEUTIL. These options will affect both the output to the screen and to the list file.

```
LIST PARAMETERS
First text line   Yes
Extra text lines  Yes
Show histories    Yes
Show debug stuff  No
Don’t skip lines  No
```

Move the highlighted box to the item you want to change (using arrow keys or **Ctrl-U**, **D**, **R**, **L**) and enter the new value (Y or N). Press **ESC** when done.

USE_FREE_STACK This command gives you access to numbers in the “free” (deleted) stack for this session only. You could then use the “C” or “D” option to copy the numbers to an ASCII file, and read them back using FONEBULD if they were mistakenly deleted.

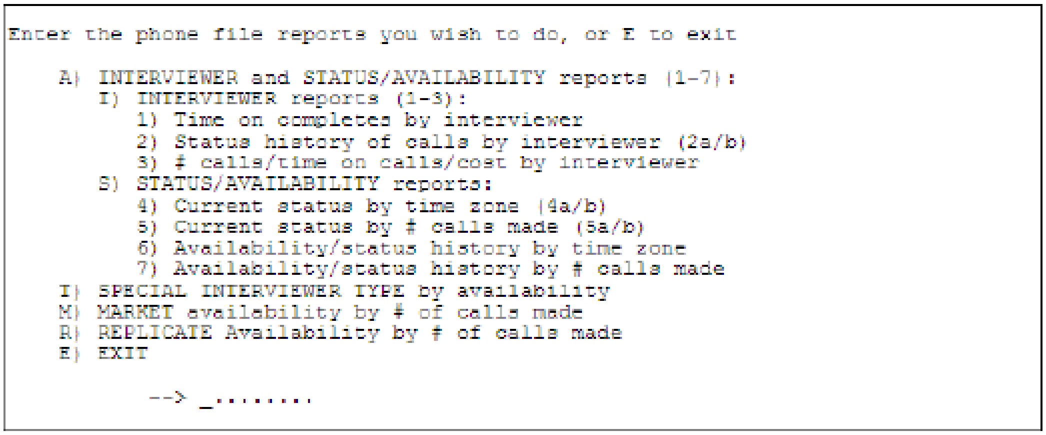
6.7.2 Using PHONERPT to Generate Status Reports

To generate status reports on your phone file, enter:

```
PHONERPT
```

You may read the FON file directly (even while the study is live), or use an ASCII file converted from a phone file by FONEUTIL. You may have multiple input files (using wildcard file references), but they must all be FON files or all be ASCII files.

You will be prompted for the study name. Then you'll be asked which reports you want:



```
Enter the phone file reports you wish to do, or E to exit

    A) INTERVIEWER and STATUS/AVAILABILITY reports (1-7):
       I) INTERVIEWER reports (1-3):
          1) Time on completes by interviewer
          2) Status history of calls by interviewer (2a/b)
          3) # calls/time on calls/cost by interviewer
       S) STATUS/AVAILABILITY reports:
          4) Current status by time zone (4a/b)
          5) Current status by # calls made (5a/b)
          6) Availability/status history by time zone
          7) Availability/status history by # calls made
    I) SPECIAL INTERVIEWER TYPE by availability
    M) MARKET availability by # of calls made
    R) REPLICATE Availability by # of calls made
    E) EXIT

           --> _........
```

Pick the appropriate code(s). Pick "A,T,M,R" to do ALL reports.

You have a choice of putting a base on your reports. This will use only selected phone records. A one-line box will be presented for you to enter your base specification. If you enter enough characters, you will be given a second box to finish your base specification in. You can also specify a start and end date and time for the reports.

For interviewer reports, you can specify whether to use the EMPLOYEE file's list of interviewer IDs, names and other information, which is faster and has the employee name and other information, or specifically generate the interviewer IDs out of the call histories in the phone file if you don't have access to the EMPLOYEE list, or the file came from some other site.

You can get cost information in the Interviewer productivity report as well (report 3). You can base the cost on either the

Amount per Interviewer Hour or the Amount per Complete. The Total cost and average cost per hour/complete will print.

You will get a standard Portrait or Landscape mode report for printing. The portrait report is useful if you are looking at the reports on a computer screen, the landscape reports have more information.

You will also get “delimited” and “html” reports generated. The delimited reports are useful to load into Excel or other spreadsheets, the html reports can be posted on a web page and seen with a browser.

This is the ONLY place to get complete MARKET, REPLICATE, and SPECIAL INTERVIEWER information as to which numbers are in which stacks.

You can customize the “status” reports by editing the file $CFMC/control/fonestat and putting the labels you want to use for each status there. These same labels will be shown to interviewers in the call histories during the interview.

Phone rpt allows for an alternate "FONESTAT" file for report labels. You can use a .fst file in the $CFMC/control directory as the label set to use for that study's phone reports. If a ".fst" file does not exist, phonerpt uses the labels in "fonestat" like before.

You can customize the report format by editing the file PHRPT^SPX in the SPX directory in the CfMC area and copying the PHRPT^DB and PHRPS^DB file to the CFMC SUPPORT directory, or you can use PHRPT^SPX to run the reports with Mentor as a batch operation.

Some of the customization features include:

1 Ability to count completes as statuses including or other than “1”.

2 Ability to set the numerator and denominator for the INCIDENCE calculation on report 5b.

Also, see example file DISPO^SPX to get a simple report of final and last statuses.

On the following pages you will see representative tables.

6.7.3 Using MAKEZONE to Build the ZONE TABLE

This program lets you add area codes to an existing ZONETABL file. Since area codes are being added so quickly in the United States, users cannot wait for each update to receive a new file. Be careful about adding area codes into the table before the area is activated, or your interviewers will be unable to successfully dial those numbers. See example file MAKEZONE^DOC for more information.

You should download a current ZONETABL file from the CfMC Support Web site and find out what the most current area codes and prefix changes have been made before running MAKEZONE to make a custom version, since the changes may already be incorporated into that version. If you do not have access to the CfMC Support Web site, contact CfMC Support to get the information to log in.

Note to predictive/preview/auto dialer users: Make sure that the dialer program has the same area codes added to its database, or the dialer may reject these numbers even though Survent thinks they're OK.

MAKEZONE reads a spec file that is an ASCII version of the ZONETABL. This spec file is the zone source file and is supplied as part of each update. The exact name of the file is operating system dependent.

\CFMC\SURVENT\ZONETABL.SRC (DOS)

/cfmc/survent/zonetabl.src (UNIX)

To add a new area code into the ZONETABL do the following:

1 Make backup copies of the current ZONETABL and zone source files. Try dialing a couple of the numbers that your current zone table is rejecting and verify that the phone rings. If it does ring, then the area code has been activated. If it doesn't ring and you get a fast busy signal, then the area code has not been activated and you probably should not add that area code into the ZONETABL. Check the current zone source file to see if the area code exists. If it does not exist, find the area ode it split off from, and if it has no exceptions, you can copy that line to the

appropriate place for the new area code and change the area code to the new area code. If the old area code has exceptions, you cannot update this area code with 100% accuracy. Contact the Support Hotline. CfMC may not have 100% accurate information for some time after the area code is activated. We must rely on our supplier to give us a list of exchanges for the new area code. If you need 100% accuracy and CfMC has not yet received updated information, then use example file FIXAC^SPX to change the old area code to the new one for each phone file.

2 After adding the area code(s) into the zone source file, run the program MAKEZONEand at the spec file put in the name of the zone source file. At the list file prompt put in a valid file name to save any error messages. If you get error messages, try to figure out what you did wrong. Most likely you did not supply a proper time zone or daylight savings setting for the area code you added.

3 If MAKEZONE runs successfully, it will create a file called ZONETABL in the local directory/group. You can then use the Mentor spec file ZONE2DOC to convert the ZONETABL back to the source file. Compare the new ZONEDOC file with the zone source file to make sure it has all your changes.

4 If ZONEDOC has all your proper changes then, you can rename the ZONETABL file into the CONTROL directory/group and rename the ZONEDOC file into the SURVENT directory/group.

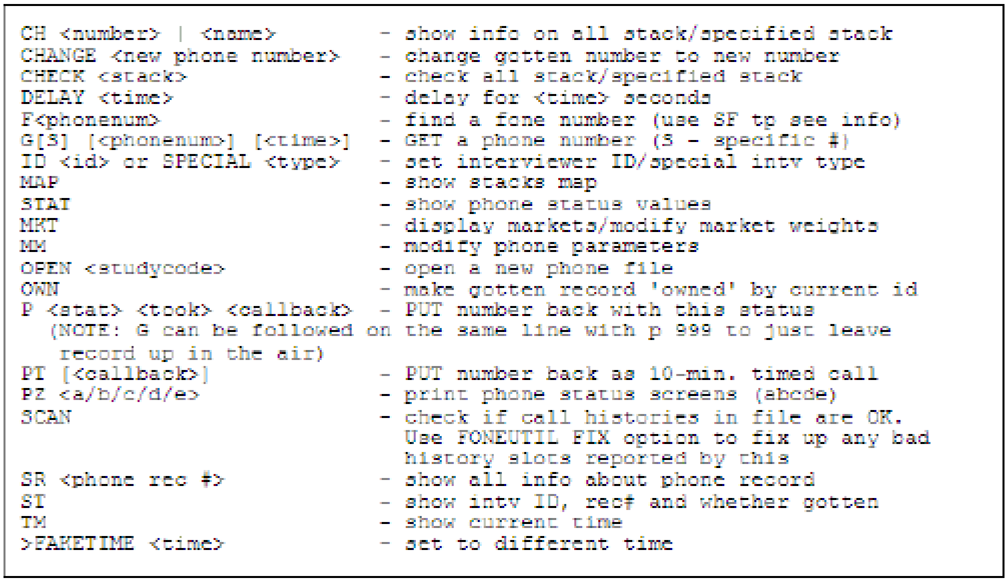
5 Use FONEBULD to add some numbers into a dummy phone file. Make sure you use a cross-section of numbers that contain both new and old area codes, plus some known bad area codes like 999. If this works, everything is OK. If it doesn't, retrace your steps to see what went wrong.

6 If you are unable to figure out what is wrong, then use your backup of the ZONETABL and contact the Support Hotline for assistance.

7 To fix just a few area codes for a particular phone file (because you need to get the study up in a hurry), see the example file FIXAC^SPX.

6.7.4 Using FONESIM to Test Phone Parameters

FONESIM stands for PHONE SYSTEM SIMULATOR. It allows you to pick up records and put them back in the file with a status, as if you were running Survent and doing interviews. It has many facilities to watch what is happening as you pick up and put down records. Many of the facilities are also available in FONEUTIL, such as modifying phone parameters and market weights, seeing status screens and so on. Here is the help screen:



```
CH <number> | <name>         - show info on all stack/specified stack
CHANGE <new phone number>    - change gotten number to new number
CHECK <stack>                - check all stack/specified stack
DELAY <time>                 - delay for <time> seconds
F<phonenum>                  - find a fone number (use SF tp see info)
G[S] [<phonenum>] [<time>]   - GET a phone number (S - specific #)
ID <id> or SPECIAL <type>    - set interviewer ID/special intv type
MAP                          - show stacks map
STAT                         - show phone status values
MKT                          - display markets/modify market weights
MM                           - modify phone parameters
OPEN <studycode>             - open a new phone file
OWN                          - make gotten record 'owned' by current id
P <stat> <took> <callback>   - PUT number back with this status
  (NOTE: G can be followed on the same line with p 999 to just leave
   record up in the air)
PT [<callback>]              - PUT number back as 10-min. timed call
PZ <a/b/c/d/e>               - print phone status screens (abcde)
SCAN                         - check if call histories in file are OK.
                               Use FONEUTIL FIX option to fix up any bad
                               history slots reported by this

SR <phone rec #>             - show all info about phone record
ST                           - show intv ID, rec# and whether gotten
TM                           - show current time
>FAKETIME <time>             - set to different time
```

The following are commonly used commands that have facilities not available in FONEUTIL:

G This gets a number. If you just say “G” it gets the next available number. If you say “GS <phonenumber>” it tries to get that phone number. If you give a time it changes the time to that time as it gets the number, so you can see which number you would get at that time.

P Puts the number back. You must specify a status. “P 101” puts the number down as a “No Answer” (see STAT for statuses).

PT <date/time> Puts the number down as a timed call (status 104) with the date/time specified, and treats it like a 10-minute call.

SCAN Checks the call history fields to make sure there is valid data there. Use the FONEUTIL “FIX” command to fix any errors.

SR Shows system fields for the record you picked up with “G”, such as call histories, record number, state code, number calls so far, etc.

>FAKETIME <date/time> Allows you to change the perceived time to try to get numbers then.

6.7.5 Command Files and Examples

CfMC provides many utilities and examples related to phone system operation. They are referenced throughout the chapter in the appropriate sections. Here is a list of the command files and utilities provided.

PHONE SYSTEM COMMAND FILES

To run the command files, you simply specify the command name and any additional parameters that are needed.

```
FONERUN <study/ALL/CURRENT/USELIST> <REBUILD/NEXTDAY>
```

This runs FONEUTIL on a phone file or set of phone files, and execute whatever options you specify in the file FNRUN^SPX. By default, it fixes and verifies the file and prints before and after information. It saves a backup copy of the file before executing. If you type “FONERUN ALL” it will do all studies in the phone directory. If you say “FONERUN USELIST” it will do all studies specified in the file USELIST. If you say “FONERUN <study>” it will operate on that study only.

Additional options are “NEXTDAY” and REBUILD”. Following is the help screen.

Type:

```
    "Fonerun <Parm1> <Parm2>"
```

```
      Parm1 must be one of the following:
ALL:         will do all files in current/fone directory
CURRENT:     will rebuild all files updated in the last 24
hours
StudyName:   will do only the phone file StudyName
USELIST:     will do all the files in the file USELIST
Parm2 can be either of the following:
NEXTDAY:     will integrate phone file as if it was tomorrow
REBUILD:     will dump the file to ASCII and rebuild it
```

This uses the file FNRUN^SPX in the local directory if there is one or it uses the one in %CFMC%SUPPORT. You can change this file to customize what FONEUTIL options will be performed on each phone file.

When the CfMCFONE variable is set, this will look for FON files in that directory, otherwise it will look in current directory. Use the CFMCENV batch file to change your environment.

```
FXRESUME <study> KILL
```

This runs FIXRESUM on a set of suspended records for a study; if the records cannot be fixed up, it runs FONEUTIL and KILLS (resolves with status 81) the phone records associated with the unfixable resume files. A backup is kept of the unresumable files.

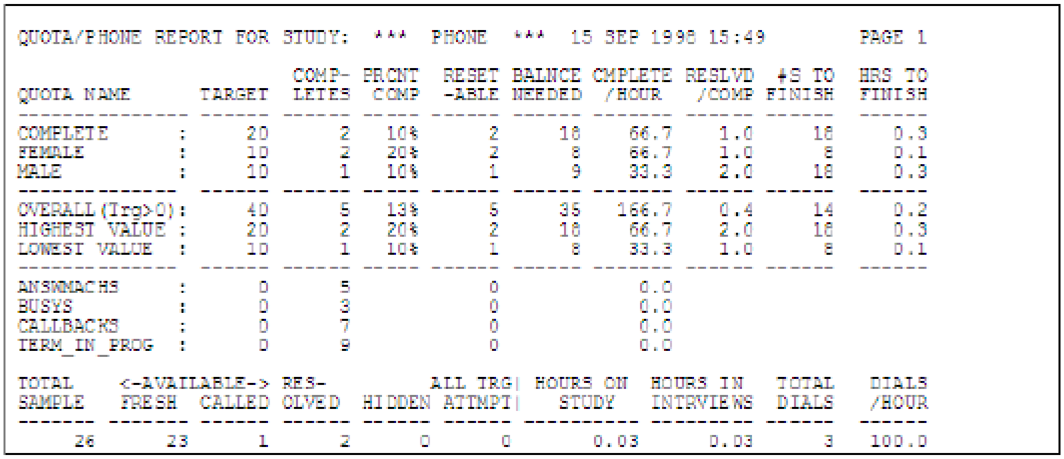
```
MAKESUSP <study> <question number>
```

This is intended to retrieve the data from suspended interviews because you wish to count the interview as a “complete” instead of continuing with the interview. It runs FONEUTIL to get a list of suspended files that were suspended after a particular question, SUSPRES to get the data from those suspends, and Mentor to combine the files. It then runs FONEUTIL to KILL (resolve with status 81) records whose data was saved.

```
QUOTARPT <study>/<study LOCAL>/ALL/USELIST/ACTIVE>
```

This utility reads study QUOTA files and FON files to produce a report which includes current quota and target counts, time on study, time to completion of quotas, and number of completes, resolved, etc.

Like FONERUN, QUOTARPT supports “ALL” to do all studies, “USELIST” to do all studies in a list, and “<study>” to do a particular study (NOTE: Use “<study> LOCAL” to get files from the local directory instead of the system area. Here is an example report:



```
QUOTA/PHONE REPORT FOR STUDY:  ***  PHONE  ***  15 SEP 1998 15:49        PAGE 1

                          COMP- PRCNT  RESET BALNCE CMPLETE RESLVD  #S TO   HRS TO
QUOTA NAME        TARGET  LETES  COMP  -ABLE NEEDED  /HOUR   /COMP FINISH   FINISH
--------------- ------ ------ ----- ------ ------ ------- ------ ------   ------
COMPLETE       :    20      2   10%      2     18    66.7    1.0     18      0.3
FEMALE         :    10      2   20%      2      8    66.7    1.0      8      0.1
MALE           :    10      1   10%      1      9    33.3    2.0     18      0.3
--------------  ------ ------ ----- ------ ------ ------- ------ ------   ------
OVERALL(Trg>0):     40      5   13%      5     35   166.7    0.4     14      0.2
HIGHEST VALUE :     20      2   20%      2     18    66.7    2.0     18      0.3
LOWEST VALUE  :     10      1   10%      1      8    33.3    1.0      8      0.1
--------------  ------ ------ ----- ------ ------ ------- ------ ------   ------
ANSWMACHS     :      0      5            0            0.0
BUSYS         :      0      3            0            0.0
CALLBACKS     :      0      7            0            0.0
TERM_IN_PROG  :      0      9            0            0.0

TOTAL    <-AVAILABLE-> RES-          ALL TRG| HOURS ON   HOURS IN  TOTAL   DIALS
SAMPLE    FRESH  CALLED OLVED  HIDDEN ATTMPT|   STUDY    INTRVIEWS DIALS   /HOUR
------- ------- ------ ------ ------ ------ ---------- --------- ------  ------
     26      23      1      2      0      0       0.03      0.03      3   100.0
```

The report tells you what you need to do to finish all of your quota groups. The numbers in the top banner come from the quota file and the numbers below from the phone file. To get all the values on the report, you must use TRIPLEQUOTAS mode (see 3.1.5 Using TRIPLEQUOTAS mode). Quotas without targets will be listed, but will not have calculations regarding percent done, balance needed, or hours to finish. Currently, this report does not report on NUMBERED quotas.

To modify the table, see the file QRPT^SPX in the CFMC SUPPORT directory.

Phone System Example Questionnaires and Mentor Spec Files

Here is an alphabetic listing of the phone system-related example files in the CFMC SURVENT directory. These are referenced throughout the chapter; use them to see exactly how different phone system features work, or just read them to see how they would be implemented.

Files with ^QPX extensions are questionnaire specifications and can generally be compiled as is or modified. Files with ^SPX extensions are Mentor spec files used to manipulate or check phone file data or produce reports. They generally require some modification of defined variables in the file to run.

This information is also available with the other Survent examples in the INDEX file in the examples area:

ADDPH^QPX shows how to add a single phone number to a phone file and use it immediately during interviewing. This is an application of the !PHONE,A statement, unlike the PHONA^QPX example, which is a questionnaire to add many records to the phone file as a separate process.

AUTOD^QPX shows one way to do autodialing using a modem in a front end screener. See MODEM^QPX for a spec file to test if a modem is working/configured properly.

BURN^QPX shows how to create a "dummy" interviewing process that does nothing other than Hide over quota numbers, so that actual interviewers are much more likely to get under quota numbers.

CHGST^SPX reads a .FON or ASCII copy of it, and will convert numbers from a resolved status and make them a timed call-back status.

CMBDP^SPX reads a .FON or ASCII copy of it, and combines information for duplicate records that were dialed into a single record. You might want to use this when converting a phone file that was built with Duplicates=Yes and to one with Duplicates=No.

DSTCV^SPX converts phone numbers not currently in the zone table to/from daylightsavings. This is for numbers not in the USA or Canada.

CNVFN^SPX converts phone files to/from version 7.6s (small) format to 7.7 (large) format.

DISPO^SPX produces a simple disposition report from the phone file showing counts of either final status for resolved numbers or last status for numbers that are still live. See PHONERPT to get full reports on the phone file.

DUPES^SPX shows how to drop duplicate numbers from a file, e.g. duplicate phone numbers. This reads the file DUPES^DAT.

EXAM2^QPX is the sample spec file used by CfMC documentation on the use of phone files. It contains a more complex contact screen than PHONE^QPX and is combined with BANK^QPX to create a more realistic questionnaire. (NOTE: You can build a sample phone file using FONEBULD and the file NUMBERS^RAW.

FIXAC^SPX reads an ASCII (raw) phone file and allows you to assign time zones to phone numbers for area codes not yet in the standard Zonetabl file used by FONEBULD to assign time zones.

FNCHK^SPX reads either a FON file or a converted ASCII phone file and reports on records that have an illegal format (corrupted phone file) and/or fixes up records that are in the error stack.

FNCNT^SPX counts the number of interviewers working on a study during each hour of the day and produces a summary report.

FNIND^QPX shows how to use the multiple index feature to get specific numbers by, for example, last name or company name. Read the file FNIND^FBL into FONEBULD to build the phone file and use the raw file of phone numbers FNIND^RAW. This uses FNIND^FBL to make the phone file.

FNEXP^SPX writes one record for every call history to be processed later. You can get all the call histories in a particular date range.

FNINT^SPX lists each call made sorted by interviewer; lists the time of the call, call status, and total time on the call.

FONEADD1 Command file to add 1 to the phone number and generate a new phone file. This is often done for RDD sample projects where you want to call the numbers that you haven’t gotten a complete with the first number you tried to call. Uses file FNAD1^SPX.

FXHST^SPX reads a FON or ASCII copy of it and will update the actual last history slot with data from the copy of the last history slot, since the last history slot is NOT used in older versions of FONEBULD when the ASCII record is read back in. This spec can also be used to modify that last history or add a history to a particular set of numbers.

MAKEZONE^DOC contains notes on how to run the MAKEZONE program so you can update your own ZONETABL file when new area codes are added. See also ZONE2^DOC, ZONESRCE and ZONET^DOC for information on the area code assignments used by MAKEZONE.

MARK^QPX shows how to set up Market areas. Use the file MARK^FBL in FONEBULD to build the file with market areas. You should always use a spec file in FONEBULD when using market areas so you can easily rebuild the file without having to re-type all the market information. Use the file MARK^RAW as the raw phone file. Uses MARK^FBL to make the phone file.

MKTCV^SPX allows you to change an existing non-market phone file into a market one, or show you how to change the current market settings in an existing market job. Markets allow you control what numbers get released by group using market weights.

MODEM^QPX builds a questionnaire to test whether a modem is configured/working properly to have the program dial the number instead of the interviewer.

MOVER^SPX reformats data from one location to another, for example, an ASCII file from an outside supplier into the format of a CfMC raw phone file.

PHLAY^QPX has a questionnaire file layout for the phone file so that you can run the REFORMAT program to export phone sample data to delimited or other layouts. Add labels and locations for your own variables to get them included with CfMC variables in the converted dataset.

PHONA^QPX shows how to add numbers to a phone file while a study is active. This has a number of different applications. See ADDPH^QPX for more information.

PHONE^CMD is a file to be read by FONEBULD to modify/create a shop file in batch mode. This can also be used to update the shopfile from release to release.

PHONE^QPX is a spec file that builds a default phone status screen and related features. This is the recommended file to start with when first attempting to use the phone system. See SHELL^QPX for a sample front/back end spec file that does not use a .FON file. See STDRD^LZW for a standardized questionnaire with more features.

PHONE^QPX is a spec file that shows how to use the Phone,L sub-type question in conjunction with markets to cause an interviewer to only dial numbers in a particular market. This can be used to control studies that are being done in multiple languages or to reduce the number of interviewers getting numbers from an almost closed market to reduce the chance of going over quota. Use the file PHONL^FBL in FONEBULD to build the fone file.

RNDFN^SPX creates one file with records from up to 10 other files, interleaving the records so there is one record from each file before the next record from the same file. This is to get all the records from separate sample files added in the order they are in the original files.

STDRD^LWZ is a set of files that let you build a phone file and write a questionnaire with standard features automatically provided, such as standard phone file positions, a standard front and back, interviewer comment area and coding and editing areas. You just provide the initial phone file layout and the body of the questionnaire, and it does the rest.

STRPF^SPX Spec file that strips unwanted characters out of the phone number field in a raw ASCII file.

UNKIL^SPX Reads a phone file or ASCIIfied phone file that was incorrectly killed and tries to undo the kill process as best as possible.

UNZAP^SPX Reads a phone file or ASCIIfied phone file that was incorrectly zapped and tries to undo the zap process as best as possible.

ZAPLS^SPX Spec file to do with commands what the "ZAP,LAST" command does in FONEUTIL. This is so you can integrate this feature into overnight applications.

6.8 AUTOMATIC PHONE DIALERS

This section includes a discussion of various issues related to installing and running CfMC's Survent in conjunction with an automatic phone dialer. The current dialers supported are a dialer from MSG PRO-T-S Systems and a predictive dialer from SER/EIS. The MSG PRO-T-S dialer can run in predictive mode or autodial mode. In autodial mode, it dials numbers for a particular station until it gets a connect; the EIS predictive dialer and the MSG PRO-T-S dialer in predictive mode dial numbers until they get a connect, then they assign the call to an available interviewer.

This section does not include information on how to install the dialers, just the interface between the dialers and the CfMC system. See the respective documentation for these systems for installation and maintenance of the dialers themselves.

PLATFORM DIFFERENCES

Directory /cfmc/ on UNIX machines is directory \CFMC\ on DOS networks. All other groups/directories follow the normal CfMC platform difference rules.

Note to UNIX Users: There are no configuration differences between UNIX systems.

6.8.1 Setting Up the Hardware and Dialer Interface

1 Before starting installation, make sure that both CfMC and dialer personnel will be available at least by phone. Also have a couple of terminals available that can be used to test to make sure that all the cabling is good. Make sure you have verified with both CfMC and dialer personnel that you have the correct versions of each software. A particular version of CfMC software (e.g., version 7.6) may only work with particular versions of EIS or MSG PRO-T-S dialer software.

IMPORTANT NOTE: Version 8.1 supports SER dialers whereever EIS was previously supported. This is due to the fact that the provider changed from EIS to SER. EIS is still supported, however. So, SER and EIS are interchangable when used in commands, etc.

2 Make sure that the dialer is running independently of CfMC and is otherwise behaving as it should be. Determine the dialer extension for each station. This may be done on an dialer by taking a single extension off hook and checking at the call processor to see which extension is off hook. ***NOTE*****:** If the correct extensions are not associated with the proper CfMC station numbers, things will not work correctly. The correspondence is maintained in the TTYINFO file or using environment variables and is described below.

3 Install the CfMC Survent software onto your system per the instructions provided.

4 Set up the interface between the computers: After configuring and connecting the ethernet cards on the computers, you should enter "ping XXXX" (where XXXX is your device name for the dialer on your network) to verify communications between the CfMC computer and dialer machine. Establish a telnet connection if possible to easily move back and forth between the machines. For DOS configurations, you must run the CfMC server on a WINDOWS-based machine with the program DIALRW.EXE running as a separate process.

5 Run the DIALCFG program to set the dialer parameters that you want to use by default. If you are not sure what you want, use the CfMC defaults to start. If you have calling codes for each study, this is where you set them and build a separate configuration file for each. Quit the program and re-run it to verify that you have properly updated the dialing parameters. You can also set the dialing parameters using commands from a file. See below for more information on how to run DIALCFG.

6 Modify the PARMFILE and TTYINFO file to describe your dialer connection(s) and options (see below).

7 Set up the terminal descriptions to include dialer extensions: If you have hard-wired ports, you set this using line descriptions in the TTYINFO file (see below). If you are connecting using TCP/IP or other non-hardwired configurations, you will need to set at least the following variables in a batch file or login script for the program to properly access the terminal. Here is an example:

UNIX:

```
setenv TERM wyse50
setenv LDEV 101
setenv EXTENSION 101
```

HP:

```
setvar TERMTYPE "wyse50"
setvar LDEV 101
setvar EXTENSION 101
```

8 If you are on a UNIX system, make sure your .login and .cshrc scripts are correct, or copy "login.exam" and "cshrc.exam" from the CfMC CONTROL directory to your login directory and adjust them accordingly. If you are running terminals under TCP/IP, set the variables as in 6) above.

9 If you are running UNIX, verify that the "atmsread" background process is not running by saying "ps -ef | grep atmsread". If it is running, kill it by typing "kill -9 <process number>".

10 Start up the CfMC “snapper” and “server” programs by entering "server". If you see error messages or the programs do not start, make sure you have the proper access to all files, the ports are properly configured, etc. Then, try again.

11 Start up a supervisor station (usually by entering "SUPER") and then preceding the commands described below with "SERVER:" to send the commands to the server.

Enter "SERVER:>DUMP s", making sure the "s" is lower-case. Enter ">DUMP show" to verify that the "s" dump switch is set. This causes dialer communication to be echoed to the screen as well as being saved in the server’s “ll<date>” log file.

Then enter "DIALER: 9 INIT", or "DIALER" then "9 INIT", where 9 is the dialer number that you are testing. This will initialize the CfMC/dialer communication and on the SERVER machine you should see some messages echoing back and forth about opening the read and write ports and that communication has been established. If you look in the log file there will be a message about the time differences between the two machines. If the machines are more than one minute apart, you could end up with some strange results in the history information. If the program hangs or does not appear to be communicating with the dialer, you need to redo the steps above.

Then enter "DIALER:9 HS:MY OWN MESSAGE". This will send the string MY OWN MESSAGE to the dialer. You should see "from atms ..." messages returned from the dialer acknowledging receipt of the message. If not, back up and try again.

12 Now, check that the CfMC system is working independent of the dialer. First, make sure that a non phone-system study works with the software. Start up a SERVER. Then logon with CfMC access at a station, and enter "NETSURV". If the program starts up, you will get a "Enter study code" prompt; enter a study code. Then you will get an interviewer ID prompt; enter a valid interviewer ID (valid IDs should be recorded in the /cfmc/cfg/employee.xxx file. If you get into a study and can collect one or more data cases, you are done.

13 Start up a SUPERVISOR (by entering "SUPER") at a terminal and make sure you can start up interviewers on a test job like example file BANK^QPX. At the interviewer station in DOS or UNIX, enter "START" or "START ###" where "###" is the device number.

14 Make sure that a phone system controlled study works independent of the dialer. Start up an interviewer on a test phone job like example file PHONE^QPX and collect some interviews.

15 Get your dialer test questionnaire ready (you may use DIALER^QPX in the CFMC Survent example directory. Make sure you have lots of phone numbers to be dialed in the sample file, and your bucket times allow calling in the time frame you are

working in. If you are using the MSG PRO-T-S dialer or preview mode using an EIS or SER dialer, make sure you have made the appropriate spec modifications to your test questionnaire (you will need a !PHONE,1 to retrieve numbers and a !PHONE,D to dial numbers). If you are using a predictive dialer such as the EIS or SER dialer, you do not need to make any spec modifications.

16 Start up one interviewer using the ATM command in the SUPERVISOR or start up "NETSURV" at a terminal and enter "<id>,ATMS" at the interviewer ID prompt. This should initialize the study for the dialer and start sending phone records from the server to the dialer,and live calls to the interviewer(s) started. The interviewers should take their phones off the hook to tell the dialer they are ready to receive numbers. They will see "...A...B...C" on their screens until the dialer connects them.

You should see various types of SHIP records on the SERVER screen (they will also be saved in the server LOG file for later review). You should see "to ATMS" messages, and "from ATMS" messages coming back. If this does not occur, either your dialer is not running or you have some configuration or hardware problem. If necessary, CfMC, MSG, or SER/EIS personnel can interpret the SHIP messages to try and isolate the problem.

17 Start as many interviewers as you can for testing and make sure the system runs with the maximum number of users running. Check that the correct phones are getting voices when the interview gets a "connect." Note: It is best to instruct the interviewers to either ask a few actual questions or ask for a person they know will not be there, so they don't disturb the test respondents.

If communications are established, and you get live voices at the correct stations, the program is configured properly and you can start your Survent study on the dialer. If not, you should take the supervisor and server down, reboot the dialer if necessary, make any necessary changes, and start your CfMC server again.

Remote phone logins with an MSG dialer

The MSG dialer allows you to call in from a remote phone to hook up to the dialer. To do this in Survent, at the interviewer ID prompt enter "rem_phone=##########".

For example:

```
Enter Interviewer ID--> mike,dialer,rem_phone=4157770470
```

The dialer will call the phone number specified and when it is picked up the interviewer will be attached to the dialer. Don't forget to set the value of CFMCEXTENSION or EXTENSION in the interviewer's login script. If you are using webCATI, you set REM_PHONE="value" on the index.html page where the interviewer logs in. You will need one index.html page for each interviewer in this case.

6.8.2 Setting Up The PARMFILE and TTYINFO file Parameters

The PARMFILE includes server parameters for the Survent software, the TTYINFO file has a description of all the terminals and devices you will be using with Survent. *See 4.4.4 INTERVIEWING CONFIGURATIONS, The TTYINFO file* for more information. The only required parameters in the PARMFILE are the TIMEZONE, EXPIRATION, and VALIDATION. Time zones are number from Greenwich Mean Time, 5=Eastern United States, 8=Western United States. The EXPIRATION and VALIDATION strings should be received from CfMC.

```
EX:
TIMEZONE: 5
```

Here are the additional parameters that are required if you are using a dialer:

DIALER: This tells Survent the type of dialer you are using. The syntax is:

```
DIALER: <EIS/GERRY/NOBLE/STRATASOFT><,dummy>
```

It is not required, but if it is missing there must be a dialer type on each DIALER## line.

If you plan to allow Survent to give special statuses to numbers that have been returned from the dialer with a disconnect status, you need to specify "<type>,DUMMY". This mode allows you to set up your own status rules by having a station signed on the study with interviewer type 9 which would receive the numbers after they were returned from the dialer to re-status them before allowing interviewers to talk to them.

DIALER##: This tells the program specific information about the dialer or dialers you are using. You may have up to 20 dialers, and you need one DIALER:## line for each to tell the program where the dialer is and what ports and extensions are involved.

```
DIALER:<type>,dummy
```

This will specify the type of any dialer that doesn't have a type on the line for that specific dialer. It is not required, in which case, every DIALER## line must have a type specified.

The syntax is:

```
DIALER##: <read><write><ext list>,<SOUND>,<baudrate> <group> <type>
      OR
DIALER## SOCKET=<IP address><read><write><ext list>,<SOUND> <group> <type>
```

<VOIP> You can use Voice-over-IP phones with the SER dialer. To do so, add "VOIP" to the dialer## line for your dialer. For example:

```
dialer08: 1995.162.1.1 5002,5003 1-240,eis,VOIP
```

This causes the SER dialer to create a connection to the VOIP phones using the extension we specify.

<IP address> is the IP address of the dialer (eg. 192,168.1.1).

<read> is the ldev or socket the dialer reads (from where it gets its input) .

<write> is the ldev or socket to which the dialer writes when it sends data to CfMC.

<ext. list> is a list of extensions connected to this dialer and it looks like <from-to>+<fromto>+<from-to>+<to> ...

<baudrate> may be present for serial connected dialers, in which case it must be 9600 or 19200. For dialers connected via sockets, this field should be blank.

<group> is an ASCII string of up to 5 characters. This will be the column head on the dialer summary screen, which you get in SURVSUPR with the "sum" command. The counts of interviewers for dialers with the same <group> will be added together.

<type> is the dialer type. If this field is missing, it will be filled in from the "dialer:" line in the parmfile. If present, it must be SER/EIS, NOBLE, GERRY or STRATASOFT.

NOTE: The fields <baudrate>, <loc>, <type> can be in any order, but you'd better not have a <loc> that looks like a <type> or a <baudrate>.

Here is an example:

```
DIALER01: SOCKET=192.183.2.2,1200,1201 41-100 mork gerry
```

"01" is the dialer number, 192.183.2.2 is the IP address for the dialer on your network (or you could use it's NAME like "DIALERX"). 1200 and 1201 are the read/write sockets for the dialer machine. 41 - 100 are the CfMC dialer extension numbers you are assigning to the phones connected to the dialer. These must be in a sequential range where 41 is the first physical dialer extension and 100 is the last (see terminal description in TTYINFO file below). Mork is the name and Gerry is the dialer type.

You can also specify disjoint extension sets. Here is an example:

```
DIALER01: 192.168.1.1 5001 5002 1-20+32+45+51-99
```

This states that the dialer 01 is at ip address 192.168.1.1, the read port is 5001, the write port is 5002, and the extensions used are 1-20, 32,45 and 51-99.

If you are using a Noble dialer, you can have Survent pass an IP address to use for the extension of the IP phone. You can set the environment variable CFMCPHONEIPADDRESS to and it will pass it along to the dialer as the address for the dialer to find for its IP phone extension.

The IP address goes to the NOBLE dialer via the Agent Login Command that CfMC sends to Noble:

```
IVR_STARTED <studycode>, <ivr ID>, <extension>,
<specialtypes>, <IP address>
```

SER/EIS_REAL_EXTENSIONS: When using multiple dialers, this causes the EIS GATEWAY displays to show the same extension numbers used by the CfMC system, instead of starting each dialer's extensions at "1" in their displays. This allows you to name interviewer stations and EIS extensions with the same number and have consistency across the CfMC and SER/EIS displays.

NOTE: The MSG PRO-T-S dialer works this way by default. (EIS_REAL_EXTENSIONS: YES)

SER/EIS_REAL_EXTENSIONS: YES causes the Portal Connect Gateway displays to show the extension numbers used by the CfMC system, instead of starting each dialer's extensions at “1” in the display. This allows you to name booths and Portal Connect extensions with the same number and have consistency across the CfMC and Portal Connect displays. This is now the default in version 7.7. (If you are running version 7.6+ in conjunction with version 7.2, set "SER/EIS_REAL_EXTENSIONS: NO")

FONENUMBER_PREFIX You can add one to six characters in front of the phone number that is passed to the dialer. You can set this on the parameter screen or use the keyword.

FONERECNUMLEN: This tells the dialer the length of the phone record number it receives from the CfMC server. The default is “5” and it may be set to “6”. Use 6 if you wish to use more than 99,999 phone records per study. This must also be set in the dialer machine configuration. (FONERECNUMLEN: 6)

CALL_MODEM_STATUS: This allows you to set a status other than 78 for a modem disconnect. In particular, you can set the status to a non-resolved status (> 100) so that the number will be called back. For more extensive control of return statuses, see the DIALER:SER/EIS,DUMMY command below. (CALL_MODEM_STATUS: 122)

CALL_INCOMPLETE_STATUS: This allows you to set a status other than 76 for an incomplete disconnect returned from the dialer. In particular, you can set the status to a non-resolved status (> 100) so that the number will be called back. For more extensive control of return statuses, see the DIALER: SER/EIS,DUMMY command above. (CALL_INCOMPLETE_STATUS: 123)

CALL_TRUNKBUSY_STATUS: ### By default, these calls were recorded with a standard “Busy” status of 102. Now, they are recorded with their own status of 157 by default. Note that statuses 157-159 now act like 102 (treated as a “busy” status). This parmfile command can be used to set the status to whatever you want. If you want it to be treated exactly like a busy, use 102 (standard "Busy" status).

The new Parmfile comand "CALL_TRUNKBUSY_STATUS: ###" can be used to set the status to whatever you want. If you want it to be treated exactly like a busy, use 102 (standard "Busy" status).

DEFAULT_USE_DIALER: YES causes any non-practice mode supervisor->interviewer startups on the system to use a dialer, unless otherwise specified. This means all supervisor "Start" and "Chi" commands will use the dialer unless "-Dialer" is specified on the command.

This is for shops where they are almost always using a dialer. It keeps the supervisors from making mistakes when starting up.

DIALER_CONNECT_STATUS: This allows you to set a status other than “1” for a connect; It would be best to use a resolved status such as 25 so if you don’t otherwise set a status it will not attempt a callback. This is useful because Survent uses status “1” as a default if you do not otherwise set a status, so there can be confusion about whether the number was a complete or never assigned a status. (DIALER_CONNECT_STATUS: 25)

DIALER_NUISANCE_STATUS: This allows you to set a status other than 106 for “nuisance” calls (numbers that were dialed by the dialer but hung up on a respondent because no interviewer was available). This would most likely be used if you did not want to call those people back; set it to a value from 2-99 to resolved the number and not call back. (DIALER_NUISANCE_STATUS: 23)

KEEP_DIALER_NUMBER:YES keeps dialer number used in call history. The keyword "KEEP_DIALER_NUMBER: YES" will cause Survent to save the dialer number used to dial the phone number into the call history as a one-column field in the first position of the usual "# seconds on call" field (6028, 6128, etc). The value returned will be "@" if there is no dialer, and letters a-t representing dialer numbers 1-20. This will also cause the "# seconds on call" field to be stored in 6029.4 instead of 6028.5, which would be used standardly.

NOTE: If you use this feature, you can only record up to "9999" seconds in the "# seconds on call" field (166 minutes).

MAX_DIALER_CLOCK_ERROR:YES indicates when to disallow dialing because the CfMC server and dialer clocks don't match. It will require that the dialer clock and the server clock agree to within the specified number of seconds or the dialer will not run.

REMOTE signals the use of "VOIP" (Voice over IP) for the SER dialer. If you add ",remote" to the extension list on the DIALER##: line in the parmfile, the software will send the commands needed to connect to "VOICE OVER IP" phones. Here is an example:

```
DIALER01: 123.212.254.012,5000,5001,20001-20200,remote
```

NOTE: In order for a supervisor to be able to voice-monitor these stations, they have to log in with an extension that is greater than 20000.

SER_INTERVIEWER_NAME:<VALUE> This controls the interviewer name sent to the SER dialer. CfMC defaults to sending the first 9 characters of the interviewer name field from the employee file (columns 5-13).

The options for "SER_INTERVIEWER_NAME:" are:

NAME	first 9 characters of the name field (5-13) (default)
ID	just the 4 character interviewer ID (1-4)

IDName	ID from 1-4, followed by a dash, followed by first part of the name (5-9)
LDevnum	The device number the interviewer is running at
Extension	The phone extension number (old default in version 7.2-)

NOTE: If you have duplicate values in that field, the program will not acknowledge the second interviewer logging on with that name. Note also that the default has changed. It used to send just the interviewer ID.

SERVER_DROP_STUDY: This tells the server to drop any study that has not been accessed for a certain number of minutes. Dialer users should be careful with this if their system is under a heavy load, as any non-connected numbers still at the dialer will be returned when the server shuts down the study. (SERVER_DROP_STUDY: 60)

6.8.3 Setting Up Device Descriptions

When using a dialer, each station you plan to use for interviewing or other CfMC purposes on your system must have a terminal type, a station number, and a dialer extension number. This needs to either be specified in the TTYINFO file, or using environment variables in the login script for the user. If you are using TELNET connections that do not have specific device numbers or TTYs, you must set environment variables to describe the devices.

Each device description includes a terminal type, station number, time zone, dialer extension, and possibly a sound channel or socket number.

Here is the syntax and an example for individual stations for UNIX:

```
Termtype TTYvalue Station# DialerExtension TimeZone SoundChannel Socket

          Example:
          wyse50 tty5a 101 11 8 15 1001
```

wyse50 is the terminal type for station 101. tty5a is the tty value for that station; this can be determined by typing tty at each individual terminal.101 defines the CfMC station number. 11 is the dialer extension and 8 is the time zone. 15 is the sound channel and 1001 is the socket to use. The last 2 values are not required.

Here is an example for individual station for DOS:

```
Station #,,TerminalType TimeZone DialerExtension SoundExtension

         Example:
         101,, WYSE50 8 11
```

101 is the device number of the Survent station. WYSE50 is the terminal type for that station. 8 is the time zone. 11 is the dialer extension.

If you are using TELNET connections, to set device numbers you must set the following environment variables as part of the user login script or batch file before issuing a start command to the device. Here is an example setting of the variables in UNIX:

```
setenv LDEV 115 (Specifies the station number)
setenv TERM wyse50 (Specifies terminal type in UNIX)
setenv CFMCPROCESSTIMEZONE 5 (Specifies the time zone
                                 for this device)
setenv EXTENSION 115 (Specifies dialer extension to use)
setenv SOUNDCHANNEL 15 (Specifies sound channel to user)
setenv SOCKET 1001 (Specifies the socket to use)
```

In UNIX under the C shell, set these using setenv <name> <value>. Use UPPER CASE for variable names and lower case for commands and values.

In DOS, use SET <name>=<value>. You do not need to set a TERM value for DOS.

6.8.4 Troubleshooting the Installation

1 Problem: Nothing is being transmitted to the DIALER machine.

- Check to be sure your read/write ports are defined correctly. If they are reversed, nothing will work.
- Hook terminals up to both the read and write ports and make sure you can logon to the host machine. Cables must be cross cables for serial port dialers (Pin 2 and Pin 3 are crossed to allow machine to talk to machine).

UNIX users note: Make sure that the read/write settings on the communications ports are set properly and that the device is disabled.

2 2) Problem: Messages are sent, but are causing strange problems like a STACK OVERFLOW or DMPT.

- Compare our log file with that of an existing known good one and make sure the messages are in the proper format.
- Compare information from our log file to the dialer logfile to verify that they both have the same information.

3 Problem: Wrong stations getting live voices.

- Check the phone wiring.
- Check that the TTYINFO FILE extension numbers match the actual phone extensions.

4 Problem: In UNIX, ' ps –ef } grep "atmsread" ' doesn't display "atmsread" process.

- Something has the wrong permissions or it doesn't exist.
- The atmsread program is missing or it is the wrong version.

6.8.5 Running The DIALCFG Program

The dialer default configuration must be set up before you can talk to the dialer. Use DIALCFG to set up the configuration file. DIALCFG will give you a screen like this:

```
dialer config v.13sep96(4,25Oct96)(,) ... HP Spectrum (C) CfMC 1978 - 1995
22 NOV 1996 11:37
Spec File-->
List File-->
Enter "OK" or "CONFIG" or “WRITE” -- >
```

```
-->
```

If you enter “OK,” this means you are done. If you type “CONFIG,” you can modify parameters. The “WRITE” command can be used when you are working interactively. After making your settings, type “WRITE” at the prompt to see the keywords and their settings. If you are running from a “Batch” file, you can use “write default” to write the default settings, or “write <study>” to write a particular study's settings. To write to a list file, you would type something like:

```
“dialcfg dialsets dialsets.out”
```

If “dialsets” contains the command “write default,” the default settings would be written to the file “dialsets.out.”

You may enter a spec file of commands or a list file to save the results, see the commands below. If you do not have commands you will press ENTER until the last prompt.

As an added option, you can put one to six characters in front of the phone number with the FONENUMBER_PREFIX command. You can set this on the parameters screen or use the keyword.

DIALCFG now can have dialer type specified on the command line or it can be entered as a new command. So you can enter:

```
Example:
DIALCFG CON CON DIALER_TYPE:GERRY
```

This can be used set the parameters for a Gerry/PRO-T-S dialer.

In addition, DIALCFG has a new "CALLER_ID" keyword for the Gerry dialer. This will be sent to the dialer after the STUDY_CONFIG message in the form "CID: ". This is the caller ID that will be specified for a project on the respondent's phone.

Main Configuration Menu

The item that you are “on” will be highlighted. To move to a new item (for UNIX users) you can either press ENTER to go to the next item, Ctrl-D to move down the screen, Ctrl-U to move up the screen, Ctrl-R to move one item to the right, or press Ctrl-L to move one item to the left. DOS users can move around the screen using the arrow keys. Once you are on an item, either you can enter an appropriate answer for it, or if it requires moving to another menu, press ESC. To return to the OK or CONFIG prompt, you must highlight the "Hit 'escape' to accept ..." line and press ESC. The main configuration menu looks like this:

```
ATMS and ATD program configuration
(Hit 'escape' to accept the configuration as it sits)
Edit the DEFAULT study configurations
Edit a NEW or EXISTING study configuration:
```

Move to the "Edit the DEFAULT ..." line and press ESC. This allows you to set up/modify the default dialing parameters that the dialer will use.

Study Configuration Menu

The modification screen will look like the following.

```
*** Configuration for study start-up parameters ***
(Hit 'escape' to accept the configuration as it sits)

This study code:
TARGET number of pending numbers per interviewer: 10
Number of NUISANCE calls allowed per 1000 dialings: 6
Number of rings (@ 6 seconds/ring) to call a 'no answer': 4
Target interviewer rest time, between calls, (in seconds): 2
Use automatic answering machine detection (0=No, 1=Yes): 0

How to treat various types of connects:
(1 - Return unconnected, 2 - Hold until study close, 3 - Connect)
Changed Number:     1      Out-Of-Service: 1
Unknown Problem:    1      Can't Complete: 1

How to treat various types of non-connects:
(1 - Return unconnected, 2 - Hold until study close)
Modem: 1 No Answer: 1 Busy: 1
Nuisance: 1 Invalid Number: 1 Answer Machine: 1
Billing code for PBX (3 - 4 digits)0 preview dialing?--> No
```

DIALCFG shows only the parameters that make sense for the dialer type(s) you have specified in the CfMC PARMFILE. For instance, the "Gerry" and "SER" dialer screens will look like this:

```
This study code:
TARGET pending per interviewer:        4
# NUISANCE calls per 1000 dialings:    3
# rings to call a 'no answer':         2
ivr rest time, between calls (sec):    4
auto ans machine detect (0=No, 1=Yes): 1

How to treat various types of connects:
(1 - Return unconnected, 3 - Connect)
Changed Number:  2       Out-Of-Service: 3
Unknown Problem: 1       Cant Complete:  1

Billing code for PBX (3 - 6 digits) preview dialing?-->No
prefix for every phone number (0 - 6 digits)-->
suffix for every phone number (0 - 6 digits)-->
```

TARGET refers to the number of numbers to send to the dialer when you start up a process.

A NUISANCE CALL is one in which a respondent is on the line but no interviewer is available. The higher the number of nuisance calls allowed, the faster the dialer will dial numbers if they are available.

TARGET INTERVIEWER REST TIME sets the average amount of time the interviewer will be able to rest between calls. This cannot be set to less than one.

HOLDING NUMBERS TO THE STUDY CLOSE can speed up the processing time, but should not otherwise be used. See the description of FLUSH below.

BILLING CODE FOR PBX is used if you need to send such a code through your phone switch before dialing a long distance number. Can be 0 (zero) if you do not need such a code. If you use different codes for each study for billing purposes, you will need to make a new configuration file for each study.

PREVIEW DIALING is used to cause the SER/EIS predictive dialer to go into preview mode. This will send a number to each individual station before it is dialed. You can cursor around setting all these options. Most of these settings can also be changed on the SER/EIS gateway, but we strongly recommend that you change them here if possible as any changes made on the SER/EIS gateway will be lost when you flush the study. Whatever settings are in this configuration are sent to the dialer when you start/restart a study.

EDIT A NEW OR EXISTING STUDY CONFIGURATION allows you to set various dialing parameters on a particular study. The study code can have three to eight characters. The study code must be entered here to either build or modify an existing study configuration.

The default configuration settings will be saved differently on the different systems:

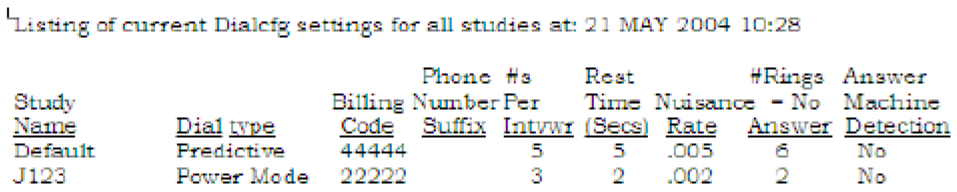
The exact name of the file is operating system-dependent.

In UNIX, the default configuration file is called ATMSCON and is saved in the CfMC CONTROL directory by default, or, if the CFMCDIALER variable is set, it puts it there. If you make a configuration for other studies, the file is called <study>.cf.

In DOS, the file is also called ATMSCON but is in the CfMC\CONTROL directory by default. The CFMCDIALER parameter will cause it to go wherever it points to. Configurations for new studies are also called <study>.cf in the same directory.

THE LISTCFG UTILITY

Once you have your study configuration files set up, you may want to check and see what setting all your studies have without having to run DIALCFG for each study. You can do this by invoking the LISTCFG utility. It lists the main settings for the default study and all studies that have been set up. Here is example output:



```
Listing of current Dialcfg settings for all studies at: 21 MAY 2004 10:28

                                      Phone  #s      Rest             #Rings  Answer
Study                         Billing Number Per     Time  Nuisance  - No    Machine
Name          Dial type       Code    Suffix Intvwr (Secs)  Rate     Answer  Detection
Default       Predictive      44444             5      5    .005       6        No
J123          Power Mode      22222             3      2    .002       2        No
```

NOTE: If your study is not listed above, it will use the 'Default' settings. As you can see, it lists the study name, type of dialing being done, billing code, phone number suffix (if any), #s released per interviewer, average rest time between interviews, nuisance rate (# per 1000), number of rings to consider a “no answer” call, and whether “answer machine detection” is used.

USING BATCH COMMANDS

You may use commands to set all the DIALCFG parameter, like other CfMC programs. This allows you to use a command file to set the parameters instead of having to work interactively. Here are the available commands:

Command	**What it does**
>batchjob	tells the program this run is using a command file
config	tells program to use commands below to configure
study: <name/DEFAULT>	set name of study config file to name or default
nums_per_intv: ##	number of numbers to assume per interviewer
nuisance: ##	number of nuisance calls allowed per 1000 dialings

number_rings: ##	number of rings to call a "no answer"
rest_time: ##	seconds allowed between interviews per interviewer
have_ans_machine: #	answering machine detection on or off
connect_changed: #	how to treat a connect for changed number (1-3)
connect_noservice: #	not in service connect
connect_unknown: #	unknown number connect
connect_traffic: #	can't complete connect
connect_modem: #	how to treat modem non-connect
connect_rna: #	no answer non-connect
connect_busy: #	busy non-connect
connect_nuisance: #	nuisance call non-connect
connect_invalid: #	invalid number non-connect
connect_ans_machine: #	answering machine connect
billingcode: ###	PBX code sent to telephone for billing purposes
Fonenumber_suffix:<characters>	Sets a suffix for the phone number when the number is sent to the dialer. (This is not to be confused with the "billing code", which is sent to the SER/EIS dialer when the study is started and is appended by SER/EIS.)
preview_mode: Y/N	whether to do preview mode
quit:	quit reading commands
OK	done with processing

To run a command file called dialcmd and list output to diallist, type:

```
EX:
dialcfg dialcmd -diallist
```

6.8.6 Dialer Issues while the Study is Live

SUPERVISOR COMMANDS AVAILABLE

There are many commands you may specify from the supervisor that are related to the dialer. Here are some of them:

ATM <interviewers>, <study> Starts interviewers on study using the dialer.

CHI <interviewers>,<study>, -DIALER Switches interviewers to study using/not using the dialer. (*See Chapter 4, section 4.4.2,* for more on this function.)

DIALER_STATUS Checks the dialers and reports whether they are running.

IS_DIALER_UP Checks the dialers and reports whether they are running.

STUDY <studyname> Returns info on study, including DUMMY number processor. For example:

```
dummy ldev = 3112 ncases = 473 nstarts = 1
Last message from dummy, time = (02 FEB 1999 12:20)
```

SUMMARY Reports how many people are on each dialer by study. Here is what a summary report would look like:

```
     Study       total dialer #1 #2 #3 #4 #5 #6 #7 #8 #9
     MA05        13     12     12
     SP05        5      2      2
     totals      18     14     2 12 .......
```

SUMMARY note: If you put a name (5 characters or less) at the end of the DIALER##: line in the parmfile, it will be displayed on the summary report instead of #1, #2, etc.

Once you have started the SERVER, you can send commands from the supervisor by preceding them with “SERVER:”

Server commands available are:

CLEARSTUDY <study> Releases study, unused numbers returned from dialer.

CLEARSTUDYNOW <study> Totally/immediately releases study from the dialer.

LOG Closes the current log file so you can look at it with an editor.

DOWN Shuts down the SERVER and returns numbers from dialer if studies are not active.

DOWNNOW Shuts down CfMC server no matter what.

DIALER:1 CLOSE Closes dialer #1 and dialer port.

DIALER:1 HANDSHAKE ## TIMES Handshakes with dialer to show activity.

DIALER:2 INIT Initializes the dialer interface on dialer #2.

DIALER:3 MSG:HS:stopatms Sends stopatms to SER/EIS dialer #3 to shut down.

DIALER:4 MSG:CC:stud:F:Y:N Closes the campaign study at SER/EIS dialer #4.

DIALER:5 WIPEOUT Closes communications with dialer no matter what

When you exit the SERVER or use the LOG command, the SERVER closes the current log file. It calls it LL######, where ###### is the day of the month followed by the military time of when that file was opened. The file is stored wherever the server is running. This is useful when debugging problems as it has a copy of all the messages that the SERVER has sent and received. This includes messages to or from Survent, SURVSUPR, SURVMON and the DIALER. This file can be used with the dialer's log file to see what happens.

SERVER AND DIALER INTERACTION

When you start the dialer on its first study in UNIX, the CfMC SERVER starts the program ATMSREAD to read messages back from the dialer. When debugging problems, make sure the ATMSREAD program is running properly. They could be incorrect versions or have bad permissions, etc.

If you are running with a dialer under DOS, the DIALRW.EXE program must also be running. This sends messages to the dialer via TCPIP, but also talks to the DOS IPCFILES in file i/o mode. DIALRW must run on the same network as the server and be available to the dialer via TCPIP.

The command line to run DIALRW is as follows:

```
DIALRW CON -XOUT LDEV:168 GERRY.CFMC.COM 8 1810 1812 231-240
```

Where:

CON is the “spec file” of commands, which in this case is “NONE”

-XOUT is the name of the list file. It contains the messages to/from the DIALRW program, which may be useful for debugging.

LDEV:168 Is the mailbox where DIALRW will receive messages from the SERVER.

GERRY.CFMC.COM Is the name of the DIALER machine on the TCPIP network.

8 Is required, and corresponds to dialer08: line in the PARMFILE

1810 1812 Are the ports the DIALRW and dialer programs read and write to.

231-240 Is the extension list, which should match the extensions in the TTYINFO file.

The log file name for DIALRW.EXE will be in this format:

```
DDHHMMSS.log
```

DD represents the day, HH, the hour, MM, the minutes, and SS, the seconds of the current month.

SER/EIS Dialer Emulation Program “DIALPS”

We now have a dialer emulation program that will run with the CfMC Server and return information from the dialer as if it was calling the numbers requested. The program name is dialps. To run it, enter:

```
dialps con -dps.lfl inport:5000 outport:5001 dialer_name:eis
```

The "dialer_name" can be “ser” or "eis", "gerry", "noble", or "strata".

You need to set the proper information in the /cfmc/control/parmfile for this to function:

```
eis_real_extensions: yes
```

```
dialer: eis
dialer01: 0.0.0.0 5000,5001 1-50+72+83-99
```

The "0.0.0.0" says to use a dummy dialer, 1-50 are the extensions that you would set to the extensions that you are using.

PREPARE Commands Related to the Dialer

When you write your questionnaire for a dialer, there are just a few issues. If you are using predictive mode, the questionnaire makes the interviewer available for calls as soon as the phone is off hook and the "!PHONE,4" statement in the questionnaire is triggered.

If you are using "preview" mode, you use the "!PHONE,1" statement to get a number from the dialer. You may then do quota checks or otherwise prepare the number to be dialed. Then you execute a "!PHONE,D" statement to dial the number. If you want to change the number to be dialed, use the syntax "!PHONE,D,<label>", where <label> has the number to call.

You can use the function "!IF DIALER()=#" to tell whether you are running in predictive or preview mode. This function returns a "1" if you are in predictive mode, a "2" if you are in preview mode, and a "0" if you are not using a dialer at all. Using this function, you can write questionnaires that allow you to run interviewers in any mode. By the way, you can have some interviewers running on the dialer and some not, but you cannot have some running in preview mode and others in predictive mode on the same study at the same time.

{!PHONE,V} and {!PHONE,W} may be used to control what happens when an interviewer goes off hook after the dialer has sent them a call. The default {!PHONE,V} is they get a message to "PICK UP THE PHONE, CALL COMING!". If you say {!PHONE,W}, a message will be sent to the dialer to hang up the call and it will get status 109 and be put back to be called later.

If you are using "Validators" in conjunction with the SER/EIS dialer (see *6.6.6 Independent Validation of Interviews Upon Completion*), you will use the !PHONE,X statement to mark the interview to be validated. This will cause the phone call to be transferred to the validating interviewer at the completion of the interview by the SER/EIS system. The CfMC system will send the phone and data record to the validator to allow the display and modification of the respondent name, etc.

6.8.7 Survent and Phone System Issues when Using a Dialer

When you start a study under the dialer, it initializes the campaign at the dialer. Then numbers are sent from the phone file in batches to the dialer depending on how many interviewers are logged on. While numbers are at the dialer, they will appear as Up in the Air in the phone status screens.

When an interviewer takes his phone off the hook, a message is sent to the dialer that they are available for calls. They will see "...A...B...C...D" until the dialer sends them a number. They could wait as much as a minute at their terminal for a number if there are not enough good numbers in the phone file. If you run out of numbers or lose contact with the server, the interviewer will get a "not getting info from server, disconnect?" message. Either release more numbers or enter "yes" to disconnect the interviewer.

You cannot display phone file information until a number is returned from the dialer. Sometimes the voice will come on the other end of the line before the screen paints if your system is slow. Make sure the interviewers have some introductory statement that will allow time for the screen to come up with the respondent's name and other information.

When a number is dialed by the dialer, it is returned to the CfMC server with a status. If it is a good connect, it is sent to an available interviewer. If the interviewers are “special” interviewers, they get numbers sent directly to them instead of through the pool of available numbers.

If a call is not connected, the number does one of three things:

1 The number is returned to the phone file where the status is recorded and rescheduled or resolved.

2 The number is given to a live interviewer because you set DIALCFG to return certain types of numbers to a live interviewer.

3 The number is given to a Dummy interviewer (interviewing session logged in with special type 9) because you have set EIS,DUMMY as the dialer type in the TTYFILE/PARMFILE. The dummy interviewer will then use questionnaire control logic to determine a special status for the number.

Here is a list of statuses returned from the dialers CFMC uses:

Name	**CfMC status**	**SER/EIS Ship Message Status**	**MSG PRO-T-S status**
CONNECTED	*1	0/K/C	CONNECTED
CALLBACK STATUSES:			
NO ANSWER	101	R	NOANSWER
BUSY	102	B	BUSY
LINE FAIL	157	T	LINEFAIL
NO LINE	157	T	NOLINE
NUISANCE CALL	106	N, d or g	ABANDON
ANSWERING MACHINE	*107	Y or y	ANSMACH
TRUNK BUSY	157	T	CIRCBUSY
NO RINGBACK	183		NORINGBACK
INCOMPLETE CALLBACK	185		
REPLACE NUMBER	211	(NOT DIALED)	
RESOLVED STATUSES:			
UNKNOWN NUMBER	73	U or u	

Name	CfMC status	SER/EIS Ship Message Status	MSG PRO-T-S status
OUT OF SERVICE	75	S or s	OOSERVICE
INCOMPLETE	*76	i	NOCOMPLT
NUMBER NOT DIALED	77	V or v	BADNUMBER
MODEM ANSWERED	*78	m	MODEM
CHANGED NUMBER	82	X or x	CHANGED
NOT DIALED	<blank>		NOTDIALED

*=status # may be modified by PARMFILE parameter

You may use the FONESTATUS( ) function when a record is returned from the dialer to find out the status returned. This is often used by the DUMMY interviewer that gets numbers back from the dialer that were not connects.

If either a supervisor (SURVSUPR) or monitor (SURVMON) wants to monitor a station, they should enter in the appropriate monitor command and then should get both the audio and video monitoring of that station if their phone is also connected to the dialer.

It is possible to set up external monitoring with a dialer that allows voice reception. In the SURVMON program, use the command "Phone xxx" where xxx is the extension you wish to use for the external voice monitoring. Contact CfMC for more details.

T A B L E   O F   C O N T E N T S
PAGE 1

```
                         Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 1

                                REPORT 1:  TOTAL TIME ON COMPLETED INTERVIEWS BY COMPLETING INTERVIEWER

                  <---------------------------- MINUTES ON (PHONE) INTERVIEW ----------------------------->
INTER-      TOTAL  LESS   ONE THREE  FIVE SEVEN  NINE    11    13    16    21    26    31    41    51  MORE    SOME  TOTAL    AVG.
VIEWER      COMP-  THAN    to    to    to    to    to    to    to    to    to    to    to    to    to  THAN    TIME  MIN-    # OF
 ID         LETES   ONE   TWO  FOUR   SIX EIGHT   TEN    12    15    20    25    30    40    50    60    60  PROBLM  UTES   MINUTES
------    ------- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ------ ------- -------

TOTAL          86     1     -     -     -     1     1     1     1     2     1     -     3     9    20    46     -    5149     59.9
            100.0%100.0%                  100.0%100.0%100.0%100.0%100.0%100.0%      100.0%100.0%100.0%100.0%        100.0%

0001            2     -     -     -     -     -     -     -     1     -     -     -     1     -     -     -     -      47     23.5
              2.3%                                          100.0%                   33.3%                            0.9%

0020           11     -     -     -     -     -     -     -     -     -     -     -     2     2     2     5     -     642     58.4
             12.8%                                                                   66.7% 22.2% 10.0% 10.9%         12.5%

0128            7     -     -     -     -     -     -     -     -     1     -     -     -     1     2     3     -     391     55.9
              8.1%                                                 50.0%                   11.1% 10.0%  6.5%          7.6%

0192            5     -     -     -     -     -     -     -     -     -     -     -     -     -     -     5     -     432     86.4
              5.8%                                                                                     10.9%          8.4%

0227            7     -     -     -     -     -     -     1     -     1     -     -     -     -     3     2     -     333     47.6
              8.1%                                    100.0%       50.0%                         15.0%  4.3%          6.5%

0260            2     -     -     -     -     -     -     -     -     -     -     -     -     -     -     2     -     157     78.5
              2.3%                                                                                      4.3%          3.0%

0267            4     -     -     -     -     -     -     -     -     -     -     -     -     -     1     3     -     262     65.5
              4.7%                                                                                5.0%  6.5%          5.1%

0272            1     -     -     -     -     -     -     -     -     -     -     -     -     -     -     1     -      84     84.0
              1.2%                                                                                      2.2%          1.6%

0284            3     -     -     -     -     -     -     -     -     -     -     -     -     -     -     3     -     220     73.3
              3.5%                                                                                      6.5%          4.3%

0285           14     -     -     -     -     -     -     -     -     -     -     -     -     2     4     8     -     887     63.4
             16.3%                                                                         22.2% 20.0% 17.4%         17.2%

0316            1     -     -     -     -     -     -     -     -     -     -     -     -     -     -     1     -      71     71.0
              1.2%                                                                                      2.2%          1.4%

0322            6     -     -     -     -     -     -     -     -     -     1     -     -     1     2     2     -     316     52.7
              7.0%                                                      100.0%             11.1% 10.0%  4.3%          6.1%

0342            8     -     -     -     -     -     -     -     -     -     -     -     -     1     1     6     -     589     73.6
              9.3%                                                                         11.1%  5.0% 13.0%         11.4%

0344           14     -     -     -     -     1     1     -     -     -     -     -     -     2     5     5     -     718     51.3
             16.3%                        100.0%100.0%                                     22.2% 25.0% 10.9%         13.9%

INTV            1     1     -     -     -     -     -     -     -     -     -     -     -     -     -     -     -       -      0.0
              1.2%100.0%
```

```
                        Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 2

                                REPORT 2A: STATUS HISTORY OF CALLS MADE BY INTERVIEWER (ACTIVE STATUSES)

                                                 TIMED   CALL- <--------- USER DEFINED CODES ----------><SPECIAL INTERVIEWER> SYSTEM
INTER-      TOTAL  TOTAL    NO            BUSY   CALL-  BACK  <--- NO ANSWERS ---><--- TIMED CALLS --->  *1*    *2*   *3-9*  (180-
VIEWER      CALLS  ACTIVE ANSWER  BUSY  ->N/A   BACK UNSPEC                (108-                (162-                 (193-   189,
 ID          MADE   STATUS (101)  (102)  (103)  (104)  (105)  (106)  (107)   159)  (160)  (161)  179)   (191)  (192)   199)   213+)
------      ------- ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------

TOTAL        12767   9482   1121   1705    426    808    236     -     886     -    2543    624    858     -      -     275     -
             100.0%  74.3%   8.8%  13.4%   3.3%   6.3%   1.8%          6.9%         19.9%   4.9%   6.7%                 2.2%
             100.0% 100.0% 100.0% 100.0% 100.0% 100.0% 100.0%        100.0%        100.0% 100.0% 100.0%               100.0%

0001             5      3      2     -      -      -      -      -      -      -      -      -      -      -      -       1     -
             100.0%  60.0%  40.0%                                                                                      20.0%
                *      *     0.2%                                                                                       0.4%

0020           479    395     43     37      9     22      3     -      51     -     123     31     60     -      -      16     -
             100.0%  82.5%   9.0%   7.7%   1.9%   4.6%   0.6%         10.6%         25.7%   6.5%  12.5%                 3.3%
               3.8%   4.2%   3.8%   2.2%   2.1%   2.7%   1.3%          5.8%          4.8%   5.0%   7.0%                 5.8%

0090            97     72      9     14      3      8      1     -       9     -      13      1     13     -      -       1     -
             100.0%  74.2%   9.3%  14.4%   3.1%   8.2%   1.0%          9.3%         13.4%   1.0%  13.4%                 1.0%
               0.8%   0.8%   0.8%   0.8%   0.7%   1.0%   0.4%          1.0%          0.5%   0.2%   1.5%                 0.4%

0101           694    521     21    149     40     27     -      -      47     -     157     19     44     -      -      17     -
             100.0%  75.1%   3.0%  21.5%   5.8%   3.9%                 6.8%         22.6%   2.7%   6.3%                 2.4%
               5.4%   5.5%   1.9%   8.7%   9.4%   3.3%                 5.3%          6.2%   3.0%   5.1%                 6.2%

0128           615    469     54     67     17     47     11     -      14     -      89      2    145     -      -      23     -
             100.0%  76.3%   8.8%  10.9%   2.8%   7.6%   1.8%          2.3%         14.5%   0.3%  23.6%                 3.7%
               4.8%   4.9%   4.8%   3.9%   4.0%   5.8%   4.7%          1.6%          3.5%   0.3%  16.9%                 8.4%

0130            13      9      2      2      1     -      -      -       1     -       3     -      -      -      -      -      -
             100.0%  69.2%  15.4%  15.4%   7.7%                        7.7%         23.1%
               0.1%   0.1%   0.2%   0.1%   0.2%                        0.1%          0.1%

0146            19     13      1      6      3     -      -      -       2     -       1     -      -      -      -      -      -
             100.0%  68.4%   5.3%  31.6%  15.8%                       10.5%          5.3%
               0.1%   0.1%   0.1%   0.4%   0.7%                        0.2%           *

0147           237    167     35     26     13     14      3     -      23     -      44      4     -      -      -       5     -
             100.0%  70.5%  14.8%  11.0%   5.5%   5.9%   1.3%          9.7%         18.6%   1.7%                        2.1%
               1.9%   1.8%   3.1%   1.5%   3.1%   1.7%   1.3%          2.6%          1.7%   0.6%                        1.8%

0192           645    515     23     67     14     85      3     -      16     -     196      4     95     -      -      12     -
             100.0%  79.8%   3.6%  10.4%   2.2%  13.2%   0.5%          2.5%         30.4%   0.6%  14.7%                 1.9%
               5.1%   5.4%   2.1%   3.9%   3.3%  10.5%   1.3%          1.8%          7.7%   0.6%  11.1%                 4.4%

0213            54     36     11      1     -      -       1     -       9     -      12      2     -      -      -      -      -
             100.0%  66.7%  20.4%   1.9%                 1.9%         16.7%         22.2%   3.7%
               0.4%   0.4%   1.0%   0.1%                 0.4%          1.0%          0.5%   0.3%

0227           422    320     51     38     15     43      6     -      32     -      73      5     51     -      -       6     -
             100.0%  75.8%  12.1%   9.0%   3.6%  10.2%   1.4%          7.6%         17.3%   1.2%  12.1%                 1.4%
               3.3%   3.4%   4.5%   2.2%   3.5%   5.3%   2.5%          3.6%          2.9%   0.8%   5.9%                 2.2%
```

```
                        Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 3

(continued)
                                   REPORT 2A: STATUS HISTORY OF CALLS MADE BY INTERVIEWER (ACTIVE STATUSES)

                                                 TIMED   CALL- <--------- USER DEFINED CODES ---------><SPECIAL INTERVIEWER> SYSTEM
INTER-      TOTAL  TOTAL    NO            BUSY   CALL-  BACK  <--- NO ANSWERS ---><--- TIMED CALLS --->  *1*    *2*   *3-9*  (180-
VIEWER      CALLS  ACTIVE ANSWER  BUSY  ->N/A   BACK  UNSPEC               (108-                (162-                 (193-   189,
 ID          MADE   STATUS (101)  (102)  (103)  (104)  (105)  (106)  (107)   159)  (160)  (161)   179)  (191)  (192)   199)   213+)
------      ------- ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------

0234           164    103     18     16      2     14      4      -      9      -     35      1      4      -      -      -      -
             100.0%  62.8%  11.0%   9.8%   1.2%   8.5%   2.4%          5.5%         21.3%   0.6%   2.4%
               1.3%   1.1%   1.6%   0.9%   0.5%   1.7%   1.7%          1.0%          1.4%   0.2%   0.5%

0238            42     35      4      6      -      3      -      -      3      -     13      6      -      -      -      -      -
             100.0%  83.3%   9.5%  14.3%          7.1%                 7.1%         31.0%  14.3%
               0.3%   0.4%   0.4%   0.4%          0.4%                 0.3%          0.5%   1.0%

0240           243    168     31     18      6      9      2      -     29      -     51      9     13      -      -      -      -
             100.0%  69.1%  12.8%   7.4%   2.5%   3.7%   0.8%         11.9%         21.0%   3.7%   5.3%
               1.9%   1.8%   2.8%   1.1%   1.4%   1.1%   0.8%          3.3%          2.0%   1.4%   1.5%

0243           186    130      9     31     10     12      6      -     15      -     34      7      3      -      -      3      -
             100.0%  69.9%   4.8%  16.7%   5.4%   6.5%   3.2%          8.1%         18.3%   3.8%   1.6%                 1.6%
               1.5%   1.4%   0.8%   1.8%   2.3%   1.5%   2.5%          1.7%          1.3%   1.1%   0.3%                 1.1%

0260           733    547     58     88     28     31      5      -     82      -    155      9     65      -      -     26      -
             100.0%  74.6%   7.9%  12.0%   3.8%   4.2%   0.7%         11.2%         21.1%   1.2%   8.9%                 3.5%
               5.7%   5.8%   5.2%   5.2%   6.6%   3.8%   2.1%          9.3%          6.1%   1.4%   7.6%                 9.5%

0267           378    309     37     57     11     46     15      -     17      -     50     52     22      -      -      2      -
             100.0%  81.7%   9.8%  15.1%   2.9%  12.2%   4.0%          4.5%         13.2%  13.8%   5.8%                 0.5%
               3.0%   3.3%   3.3%   3.3%   2.6%   5.7%   6.4%          1.9%          2.0%   8.3%   2.6%                 0.7%

0268            42     26      -      6      2      4      -      -      -      -      7      1      6      -      -      -      -
             100.0%  61.9%         14.3%   4.8%   9.5%                               16.7%   2.4%  14.3%
               0.3%   0.3%          0.4%   0.5%   0.5%                                0.3%   0.2%   0.7%

0272           262    207     41     26      7     30      -      -     11      -     48      6     28      -      -     10      -
             100.0%  79.0%  15.6%   9.9%   2.7%  11.5%                 4.2%         18.3%   2.3%  10.7%                 3.8%
               2.1%   2.2%   3.7%   1.5%   1.6%   3.7%                 1.2%          1.9%   1.0%   3.3%                 3.6%

0282           339    219     29     34      9     10      6      -     31      -     71     26      3      -      -      -      -
             100.0%  64.6%   8.6%  10.0%   2.7%   2.9%   1.8%          9.1%         20.9%   7.7%   0.9%
               2.7%   2.3%   2.6%   2.0%   2.1%   1.2%   2.5%          3.5%          2.8%   4.2%   0.3%

0284           512    363     56     81     18     18      2      -     21      -    138     14     14      -      -      1      -
             100.0%  70.9%  10.9%  15.8%   3.5%   3.5%   0.4%          4.1%         27.0%   2.7%   2.7%                 0.2%
               4.0%   3.8%   5.0%   4.8%   4.2%   2.2%   0.8%          2.4%          5.4%   2.2%   1.6%                 0.4%

0285          1174    797     89    155     38     91     20      -    103      -    245      6     19      -      -     31      -
             100.0%  67.9%   7.6%  13.2%   3.2%   7.8%   1.7%          8.8%         20.9%   0.5%   1.6%                 2.6%
               9.2%   8.4%   7.9%   9.1%   8.9%  11.3%   8.5%         11.6%          9.6%   1.0%   2.2%                11.3%

0295            19     16      1      3      2      -      2      -      2      -      4      1      -      -      -      1      -
             100.0%  84.2%   5.3%  15.8%  10.5%         10.5%         10.5%         21.1%   5.3%                        5.3%
               0.1%   0.2%   0.1%   0.2%   0.5%          0.8%          0.2%          0.2%   0.2%                        0.4%
```

```
                        Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 4

(continued)
                               REPORT 2A: STATUS HISTORY OF CALLS MADE BY INTERVIEWER (ACTIVE STATUSES)

                                                 TIMED   CALL- <--------- USER DEFINED CODES ----------><SPECIAL INTERVIEWER> SYSTEM
INTER-      TOTAL  TOTAL    NO            BUSY   CALL-  BACK  <--- NO ANSWERS ---><--- TIMED CALLS --->  *1*    *2*   *3-9*  (180-
VIEWER      CALLS  ACTIVE ANSWER  BUSY  ->N/A  BACK  UNSPEC                (108-               (162-                 (193-   189,
 ID         MADE    STATUS (101)  (102)  (103)  (104)  (105)  (106)  (107)   159)  (160)  (161)  179)   (191)  (192)   199)   213+)
------     ------- ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------

0316          724    540     68    102     20     45     12      -     64      -    153     12     55      -      -      9      -
           100.0%  74.6%   9.4%  14.1%   2.8%   6.2%   1.7%          8.8%         21.1%   1.7%   7.6%                 1.2%
             5.7%   5.7%   6.1%   6.0%   4.7%   5.6%   5.1%          7.2%          6.0%   1.9%   6.4%                 3.3%

0320          104     81      6      7      3      4      -      -     11      -     33      7      8      -      -      2      -
           100.0%  77.9%   5.8%   6.7%   2.9%   3.8%                10.6%         31.7%   6.7%   7.7%                 1.9%
             0.8%   0.9%   0.5%   0.4%   0.7%   0.5%                 1.2%          1.3%   1.1%   0.9%                 0.7%

0321          297    215     25     43     10      1     36      -      9      -     77      -     12      -      -      2      -
           100.0%  72.4%   8.4%  14.5%   3.4%   0.3%  12.1%          3.0%         25.9%          4.0%                 0.7%
             2.3%   2.3%   2.2%   2.5%   2.3%   0.1%  15.3%          1.0%          3.0%          1.4%                 0.7%

0322          465    373     67     58     19     36      2      -     47      -     89      2     33      -      -     20      -
           100.0%  80.2%  14.4%  12.5%   4.1%   7.7%   0.4%         10.1%         19.1%   0.4%   7.1%                 4.3%
             3.6%   3.9%   6.0%   3.4%   4.5%   4.5%   0.8%          5.3%          3.5%   0.3%   3.8%                 7.3%

0334           21     16      1      4      -      2      -      -      1      -      4      -      3      -      -      1      -
           100.0%  76.2%   4.8%  19.0%          9.5%                 4.8%         19.0%         14.3%                 4.8%
             0.2%   0.2%   0.1%   0.2%          0.2%                 0.1%          0.2%          0.3%                 0.4%

0342          686    540     78     76     16      6     50      -     36      -     58    180     16      -      -     24      -
           100.0%  78.7%  11.4%  11.1%   2.3%   0.9%   7.3%          5.2%          8.5%  26.2%   2.3%                 3.5%
             5.4%   5.7%   7.0%   4.5%   3.8%   0.7%  21.2%          4.1%          2.3%  28.8%   1.9%                 8.7%

0343         1070    687     58    221     53      1     27      -     23      -    162    132      -      -      -     10      -
           100.0%  64.2%   5.4%  20.7%   5.0%   0.1%   2.5%          2.1%         15.1%  12.3%                        0.9%
             8.4%   7.2%   5.2%  13.0%  12.4%   0.1%  11.4%          2.6%          6.4%  21.2%                        3.6%

0344         1362   1048    109    198     39    169     10      -     93      -    261     56     75      -      -     38      -
           100.0%  76.9%   8.0%  14.5%   2.9%  12.4%   0.7%          6.8%         19.2%   4.1%   5.5%                 2.8%
            10.7%  11.1%   9.7%  11.6%   9.2%  20.9%   4.2%         10.5%         10.3%   9.0%   8.7%                13.8%

0349           16     14      1      2      -      -      -      -      -      -      8      1      2      -      -      -      -
           100.0%  87.5%   6.3%  12.5%                                              50.0%   6.3%  12.5%
             0.1%   0.1%   0.1%   0.1%                                               0.3%   0.2%   0.2%

0350          145    132     27      7      6      8      -      -     16      -     30      8     23      -      -      7      -
           100.0%  91.0%  18.6%   4.8%   4.1%   5.5%                11.0%         20.7%   5.5%  15.9%                 4.8%
             1.1%   1.4%   2.4%   0.4%   1.4%   1.0%                 1.8%          1.2%   1.3%   2.7%                 2.5%

0351          217    172     24     20      3     12      1      -     21      -     42      9     39      -      -      1      -
           100.0%  79.3%  11.1%   9.2%   1.4%   5.5%   0.5%          9.7%         19.4%   4.1%  18.0%                 0.5%
             1.7%   1.8%   2.1%   1.2%   0.7%   1.5%   0.4%          2.4%          1.7%   1.4%   4.5%                 0.4%

0353          283    222     32     39      9     10      8      -     38      -     64     11      7      -      -      4      -
           100.0%  78.4%  11.3%  13.8%   3.2%   3.5%   2.8%         13.4%         22.6%   3.9%   2.5%                 1.4%
             2.2%   2.3%   2.9%   2.3%   2.1%   1.2%   3.4%          4.3%          2.5%   1.8%   0.8%                 1.5%
```

```
                        Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 5

(continued)
                               REPORT 2A: STATUS HISTORY OF CALLS MADE BY INTERVIEWER (ACTIVE STATUSES)

                                                 TIMED   CALL- <--------- USER DEFINED CODES ---------><SPECIAL INTERVIEWER> SYSTEM
INTER-      TOTAL  TOTAL    NO            BUSY   CALL-  BACK  <--- NO ANSWERS ---><--- TIMED CALLS ---> *1*    *2*   *3-9*  (180-
VIEWER      CALLS  ACTIVE ANSWER  BUSY  ->N/A  BACK  UNSPEC                (108-               (162-                 (193-   189,
  ID        MADE    STATUS (101)  (102)  (103)  (104)  (105)  (106)  (107)   159)  (160)  (161)  179)   (191)  (192)   199)   213+)
------     ------- ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------

INTV            3      2      -      -      -      -      -      -      -      -      -      -      -      -      -      2      -
            100.0%  66.7%                                                                                             66.7%
                *      *                                                                                               0.7%
```

Note: Percentage less than 0.05 printed as *.

```
Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 6

REPORT 2B: LAST STATUS OF CALLS MADE BY INTERVIEWER (RESOLVED CALLS)

                                           LANG-  TERMI   NON-   NON-    NON- <---------- TERMINATES --------->    USER-DEFINED
INTER-       TOTAL  COMP-  OTHER    RE-  UAGE  -NATED WORKNG  RESI-   BUSI-  TERM.  TERM.  TERM.  TERM.  TERM. <---- STATUSES ---->
VIEWER     RESOLVD  LETES RESOLVD FUSED PROBLM INTRVW NUMBER DENTAL   NESS   A      B      C      D      E                      (15-
 ID          CALLS   (1)  (2-93)   (2)    (3)    (4)    (5)    (6)    (7)    (8)    (9)   (10)   (11)   (12)   (13)   (14)   (93)
------     ------- ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------

TOTAL         3259     86   3017   1191     14      5   1354     -      -     126     61     -      -      -      -      -     422
            100.0%   2.6%  92.6%  36.5%   0.4%   0.2%  41.5%                 3.9%   1.9%                                   12.9%
            100.0% 100.0% 100.0% 100.0% 100.0% 100.0% 100.0%               100.0% 100.0%                                  100.0%

0001             2      2     -      -      -      -      -      -      -      -      -      -      -      -      -      -      -
            100.0% 100.0%
              0.1%   2.3%

0020            84     11     71     14     -       2     42     -      -       1      2     -      -      -      -      -      12
            100.0%  13.1%  84.5%  16.7%          2.4%  50.0%                 1.2%   2.4%                                   14.3%
              2.6%  12.8%   2.4%   1.2%         40.0%   3.1%                 0.8%   3.3%                                    2.8%

0090            25     -      22     11     -      -       8     -      -       2     -      -      -      -      -      -       4
            100.0%         88.0%  44.0%                32.0%                 8.0%                                          16.0%
              0.8%          0.7%   0.9%                 0.6%                 1.6%                                           0.9%

0101           171     -     167     93     -       1     62     -      -       4      2     -      -      -      -      -       9
            100.0%         97.7%  54.4%          0.6%  36.3%                 2.3%   1.2%                                    5.3%
              5.2%          5.5%   7.8%         20.0%   4.6%                 3.2%   3.3%                                    2.1%

0128           142      7    123     33      1     -      74     -      -       3      6     -      -      -      -      -      18
            100.0%   4.9%  86.6%  23.2%   0.7%         52.1%                 2.1%   4.2%                                   12.7%
              4.4%   8.1%   4.1%   2.8%   7.1%          5.5%                 2.4%   9.8%                                    4.3%

0130             4     -       3      1     -      -       2     -      -      -      -      -      -      -      -      -       1
            100.0%         75.0%  25.0%                50.0%                                                               25.0%
              0.1%          0.1%   0.1%                 0.1%                                                                0.2%

0146             6     -       6      2     -      -       4     -      -      -      -      -      -      -      -      -      -
            100.0%        100.0%  33.3%                66.7%
              0.2%          0.2%   0.2%                 0.3%

0147            70     -      66     29     -      -      26     -      -       4      4     -      -      -      -      -       7
            100.0%         94.3%  41.4%                37.1%                 5.7%   5.7%                                   10.0%
              2.1%          2.2%   2.4%                 1.9%                 3.2%   6.6%                                    1.7%

0192           130      5    119     59      1     -      40     -      -      -       9     -      -      -      -      -      16
            100.0%   3.8%  91.5%  45.4%   0.8%         30.8%                        6.9%                                   12.3%
              4.0%   5.8%   3.9%   5.0%   7.1%          3.0%                       14.8%                                    3.8%

0213            18     -      17      2     -      -      10     -      -      -       1     -      -      -      -      -       5
            100.0%         94.4%  11.1%                55.6%                        5.6%                                   27.8%
              0.6%          0.6%   0.2%                 0.7%                        1.6%                                    1.2%

0227            99      7     88     14     -      -      58     -      -      -       1     -      -      -      -      -      19
            100.0%   7.1%  88.9%  14.1%                58.6%                        1.0%                                   19.2%
              3.0%   8.1%   2.9%   1.2%                 4.3%                        1.6%                                    4.5%
```

```
(continued)
                        REPORT 2B: LAST STATUS OF CALLS MADE BY INTERVIEWER (RESOLVED CALLS)

                                          LANG-   TERMI  NON-   NON-    NON-  <---------  TERMINATES  --------->     USER-DEFINED
INTER-         TOTAL  COMP-  OTHER    RE-  UAGE  -NATED WORKNG RESI-   BUSI-  TERM.  TERM.  TERM.  TERM.  TERM. <---- STATUSES ---->
VIEWER       RESOLVD  LETES RESOLVD FUSED PROBLM INTRVW NUMBER DENTAL NESS   A      B      C      D      E                    (15-
 ID            CALLS    (1)  (2-93)   (2)    (3)    (4)    (5)    (6)    (7)    (8)    (9)   (10)   (11)   (12)   (13)   (14)   (93)
------       ------- ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------

0234             61      -     60     12      -      -     40      -      -      2      -      -      -      -      -      -      7
             100.0%         98.4%  19.7%                65.6%                 3.3%                                          11.5%
               1.9%          2.0%   1.0%                 3.0%                 1.6%                                           1.7%

0238              7      -      7      5      -      -      -      -      -      -      -      -      -      -      -      -      2
             100.0%        100.0%  71.4%                                                                                    28.6%
               0.2%          0.2%   0.4%                                                                                     0.5%

0240             75      -     73     24      -      -     34      -      -     10      4      -      -      -      -      -      3
             100.0%         97.3%  32.0%                45.3%                13.3%   5.3%                                    4.0%
               2.3%          2.4%   2.0%                 2.5%                 7.9%   6.6%                                    0.7%

0243             56      -     55     12      -      -     28      -      -      6      -      -      -      -      -      -     10
             100.0%         98.2%  21.4%                50.0%                10.7%                                          17.9%
               1.7%          1.8%   1.0%                 2.1%                 4.8%                                           2.4%

0260            184      2    174     85      -      -     59      -      -      -      2      -      -      -      -      -     36
             100.0%   1.1%  94.6%  46.2%                32.1%                        1.1%                                   19.6%
               5.6%   2.3%   5.8%   7.1%                 4.4%                        3.3%                                    8.5%

0267             67      4     58     16      -      -     23      -      -      4      4      -      -      -      -      -     16
             100.0%   6.0%  86.6%  23.9%                34.3%                 6.0%   6.0%                                   23.9%
               2.1%   4.7%   1.9%   1.3%                 1.7%                 3.2%   6.6%                                    3.8%

0268             16      -     16      3      -      -      2      -      -     10      1      -      -      -      -      -      -
             100.0%        100.0%  18.8%                12.5%                62.5%   6.3%
               0.5%          0.5%   0.3%                 0.1%                 7.9%   1.6%

0272             55      1     49     23      1      -     17      -      -      -      3      -      -      -      -      -     10
             100.0%   1.8%  89.1%  41.8%   1.8%         30.9%                        5.5%                                   18.2%
               1.7%   1.2%   1.6%   1.9%   7.1%          1.3%                        4.9%                                    2.4%

0282            120      -    117     53      -      -     42      -      -     19      1      -      -      -      -      -      5
             100.0%         97.5%  44.2%                35.0%                15.8%   0.8%                                    4.2%
               3.7%          3.9%   4.5%                 3.1%                15.1%   1.6%                                    1.2%

0284            149      3    137     60      2      -     55      -      -      1      3      -      -      -      -      -     25
             100.0%   2.0%  91.9%  40.3%   1.3%         36.9%                 0.7%   2.0%                                   16.8%
               4.6%   3.5%   4.5%   5.0%  14.3%          4.1%                 0.8%   4.9%                                    5.9%

0285            375     14    343    169      2      1    118      -      -     36      3      -      -      -      -      -     32
             100.0%   3.7%  91.5%  45.1%   0.5%   0.3%  31.5%                 9.6%   0.8%                                    8.5%
              11.5%  16.3%  11.4%  14.2%  14.3%  20.0%   8.7%                28.6%   4.9%                                    7.6%

0295              3      -      2      2      -      -      -      -      -      -      -      -      -      -      -      -      1
             100.0%         66.7%  66.7%                                                                                    33.3%
               0.1%          0.1%   0.2%                                                                                     0.2%
```

```
                        Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 8

(continued)
                                  REPORT 2B: LAST STATUS OF CALLS MADE BY INTERVIEWER (RESOLVED CALLS)

                                          LANG-   TERMI  NON-   NON-   NON-  <---------  TERMINATES  --------->    USER-DEFINED
INTER-       TOTAL   COMP-  OTHER   RE-  UAGE   -NATED WORKNG RESI-   BUSI-  TERM.  TERM.  TERM.  TERM.  TERM. <---- STATUSES ---->
VIEWER     RESOLVD  LETES RESOLVD FUSED PROBLM INTRVW NUMBER DENTAL NESS     A      B      C      D      E                    (15-
 ID          CALLS    (1)  (2-93)  (2)    (3)    (4)    (5)    (6)    (7)    (8)    (9)    (10)   (11)   (12)   (13)   (14)   (93)
------      ------- ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------

0316           182      1    168     48      -      -    100      -      -      -      3      -      -      -      -      -     30
             100.0%   0.5%  92.3%  26.4%                54.9%                       1.6%                                   16.5%
               5.6%   1.2%   5.6%   4.0%                 7.4%                       4.9%                                    7.1%

0320            23      -     23     12      -      -      4      -      -      -      -      -      -      -      -      -      7
             100.0%        100.0%  52.2%                17.4%                                                              30.4%
               0.7%          0.8%   1.0%                 0.3%                                                               1.7%

0321            82      -     79     23      -      -     52      -      -      -      -      -      -      -      -      -      7
             100.0%         96.3%  28.0%                63.4%                                                               8.5%
               2.5%          2.6%   1.9%                 3.8%                                                               1.7%

0322            91      6     79     35      -      -     43      -      -      -      -      -      -      -      -      -      7
             100.0%   6.6%  86.8%  38.5%                47.3%                                                               7.7%
               2.8%   7.0%   2.6%   2.9%                 3.2%                                                               1.7%

0334             5      -      4      1      -      -      1      -      -      -      -      -      -      -      -      -      3
             100.0%         80.0%  20.0%                20.0%                                                              60.0%
               0.2%          0.1%   0.1%                 0.1%                                                               0.7%

0342           146      8    138     36      -      -     88      -      -      4      -      -      -      -      -      -     10
             100.0%   5.5%  94.5%  24.7%                60.3%                2.7%                                           6.8%
               4.5%   9.3%   4.6%   3.0%                 6.5%                3.2%                                           2.4%

0343           379      -    371    192      6      1    145      -      -      5      -      -      -      -      -      -     30
             100.0%         97.9%  50.7%   1.6%   0.3%  38.3%                1.3%                                           7.9%
              11.6%         12.3%  16.1%  42.9%  20.0%  10.7%                4.0%                                           7.1%

0344           312     14    270     84      1      -    129      -      -      -     11      -      -      -      -      -     73
             100.0%   4.5%  86.5%  26.9%   0.3%         41.3%                       3.5%                                   23.4%
               9.6%  16.3%   8.9%   7.1%   7.1%          9.5%                      18.0%                                   17.3%

0349             2      -      2      1      -      -      1      -      -      -      -      -      -      -      -      -      -
             100.0%        100.0%  50.0%                50.0%
               0.1%          0.1%   0.1%                 0.1%

0350            13      -     13      -      -      -      8      -      -      2      1      -      -      -      -      -      2
             100.0%        100.0%                       61.5%               15.4%   7.7%                                   15.4%
               0.4%          0.4%                        0.6%                1.6%   1.6%                                    0.5%

0351            44      -     42     21      -      -      4      -      -     10      -      -      -      -      -      -      9
             100.0%         95.5%  47.7%                 9.1%               22.7%                                          20.5%
               1.4%          1.4%   1.8%                 0.3%                7.9%                                           2.1%

0353            60      -     55     16      -      -     35      -      -      3      -      -      -      -      -      -      6
             100.0%         91.7%  26.7%                58.3%                5.0%                                          10.0%
               1.8%          1.8%   1.3%                 2.6%                2.4%                                           1.4%
```

```
                        Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 9

(continued)
                                 REPORT 2B: LAST STATUS OF CALLS MADE BY INTERVIEWER (RESOLVED CALLS)

                                         LANG-   TERMI  NON-  NON-    NON-  <---------  TERMINATES --------->     USER-DEFINED
INTER-      TOTAL   COMP-  OTHER   RE-  UAGE   -NATED WORKNG RESI-   BUSI-  TERM.  TERM.  TERM.  TERM.  TERM. <---- STATUSES ---->
VIEWER    RESOLVD  LETES RESOLVD FUSED PROBLM INTRVW NUMBER DENTAL NESS     A      B      C      D      E                    (15-
 ID         CALLS    (1)  (2-93)  (2)    (3)    (4)    (5)    (6)    (7)    (8)    (9)    (10)   (11)   (12)   (13)   (14)   (93)
------     ------- ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------

INTV            1      1      -      -      -      -      -      -      -      -      -      -      -      -      -      -      -
            100.0% 100.0%
                *    1.2%

Note: Percentage less than 0.05 printed as *.
```

```
                         Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 10

                                               REPORT 3:  INTERVIEWER PRODUCTIVITY REPORT

                         C A L L S    M A D E                  H O U R S    L O G G E D                    S U M M A R Y
            <------------------------------------------> <----------------------------------------> <---------------------------------->
INTER-       TOTAL  COMP-  OTHER  CALL-  BUSY/  OTHER    TOTAL   COMP- SUSPEND  OTHER     BUSY,    CALLS   COMP-       TOTAL    COST
VIEWER       CALLS  LETED  RESOL  BACKS     NO ACTIVE    PHONE   LETED ED INT-  RESOL      N/A,     PER    LETES    INTRVWER  /COMP-
 ID          MADE  INTVWS   -VED  SCHLD ANSWER STATUS    HOURS INTRVWS ERVIEWS   -VED     OTHER    HOUR   /HOUR        COST    LETE
------      ------- ------ -----  ----- ------ ------  ------- ------- ------- ------- -------  ------   -----    --------  ------

TOTAL        12767     91   3194   4833   4374    275    492.3    69.7       -    74.2    348.4      25.9     0.2   $9,845.33 $108.19
            100.0%   0.7%  25.0%  37.9%  34.3%   2.2%   100.0%   14.2%           15.1%    70.8%
            100.0% 100.0% 100.0% 100.0% 100.0% 100.0%   100.0%  100.0%          100.0%   100.0%

0001             5      2      -      -      2      1      0.7     0.6       -       -      0.0       7.7     3.1      $13.00   $6.50
            100.0%  40.0%                40.0%  20.0%   100.0%   94.9%                      5.1%
                 *   2.2%                    *   0.4%     0.1%    0.9%                        *

0020           479     11     73    236    143     16     26.8     8.5       -     2.0     16.4      17.9     0.4     $536.00  $48.73
            100.0%   2.3%  15.2%  49.3%  29.9%   3.3%   100.0%   31.5%            7.5%    61.0%
              3.8%  12.1%   2.3%   4.9%   3.3%   5.8%     5.4%   12.1%            2.7%     4.7%

0090            97      -     25     35     36      1      3.6       -       -     0.5      3.1      26.8       -      $72.33       -
            100.0%         25.8%  36.1%  37.1%   1.0%   100.0%                   14.7%    85.3%
              0.8%          0.8%   0.7%   0.8%   0.4%     0.7%                    0.7%     0.9%

0101           694      -    173    247    257     17     22.8       -       -     4.5     18.3      30.5       -     $455.00       -
            100.0%         24.9%  35.6%  37.0%   2.4%   100.0%                   19.6%    80.4%
              5.4%          5.4%   5.1%   5.9%   6.2%     4.6%                    6.0%     5.3%

0128           615     10    136    283    163     23     23.1     5.2       -     2.5     15.4      26.7     0.4     $461.00  $46.10
            100.0%   1.6%  22.1%  46.0%  26.5%   3.7%   100.0%   22.6%           10.8%    66.6%
              4.8%  11.0%   4.3%   5.9%   3.7%   8.4%     4.7%    7.5%            3.3%     4.4%

0130            13      -      4      3      6      -      0.2       -       -     0.1      0.2      55.7       -       $4.67       -
            100.0%         30.8%  23.1%  46.2%          100.0%                   28.6%    71.4%
              0.1%          0.1%   0.1%   0.1%                 *                  0.1%        *

0146            19      -      6      1     12      -      0.9       -       -     0.2      0.7      21.9       -      $17.33       -
            100.0%         31.6%   5.3%  63.2%          100.0%                   23.1%    76.9%
              0.1%          0.2%      *   0.3%            0.2%                    0.3%     0.2%

0147           237      -     70     62    100      5      5.5       -       -     1.1      4.5      43.0       -     $110.33       -
            100.0%         29.5%  26.2%  42.2%   2.1%   100.0%                   19.3%    80.7%
              1.9%          2.2%   1.3%   2.3%   1.8%     1.1%                    1.4%     1.3%

0192           645      5    125    380    123     12     33.0     6.0       -     3.5     23.5      19.5     0.2     $660.00 $132.00
            100.0%   0.8%  19.4%  58.9%  19.1%   1.9%   100.0%   18.2%           10.5%    71.3%
              5.1%   5.5%   3.9%   7.9%   2.8%   4.4%     6.7%    8.6%            4.7%     6.8%

0213            54      -     18     14     22      -      1.6       -       -     0.3      1.3      34.8       -      $31.00       -
            100.0%         33.3%  25.9%  40.7%          100.0%                   18.3%    81.7%
              0.4%          0.6%   0.3%   0.5%            0.3%                    0.4%     0.4%

0227           422      7     95    172    142      6     17.0     4.0       -     1.8     11.2      24.8     0.4     $340.00  $48.57
            100.0%   1.7%  22.5%  40.8%  33.6%   1.4%   100.0%   23.5%           10.6%    65.9%
              3.3%   7.7%   3.0%   3.6%   3.2%   2.2%     3.5%    5.7%            2.4%     3.2%
```

```
(continued)
                                                   REPORT 3:  INTERVIEWER PRODUCTIVITY REPORT

                         C A L L S   M A D E                   H O U R S   L O G G E D                 S U M M A R Y
            <-------------------------------------------> <-----------------------------------------> <---------------------------------->
INTER-       TOTAL   COMP-  OTHER  CALL-  BUSY/  OTHER    TOTAL   COMP-  SUSPEND  OTHER   BUSY,    CALLS   COMP-     TOTAL    COST
VIEWER       CALLS   LETED  RESOL  BACKS     NO ACTIVE    PHONE   LETED  ED INT-  RESOL    N/A,     PER   LETES   INTRVWER  /COMP-
 ID          MADE   INTVWS   -VED  SCHLD ANSWER STATUS    HOURS  INTRVWS ERVIEWS   -VED   OTHER    HOUR   /HOUR     COST     LETE
------      ------- ------ -----  ----- ------ ------   ------- ------- -------  ------- -------  ------  -----   --------  ------

0234          164      -     61     54     49      -       4.0       -       -      1.1     2.9    41.5      -     $79.00        -
            100.0%         37.2%  32.9%  29.9%           100.0%                   27.0%   73.0%
              1.3%          1.9%   1.1%   1.1%             0.8%                    1.4%    0.8%

0238           42      -      7     22     13      -       1.0       -       -      0.1     0.9    42.0      -     $20.00        -
            100.0%         16.7%  52.4%  31.0%           100.0%                    8.3%   91.7%
              0.3%          0.2%   0.5%   0.3%             0.2%                    0.1%    0.3%

0240          243      -     75     82     86      -       7.1       -       -      1.5     5.6    34.2      -    $142.00        -
            100.0%         30.9%  33.7%  35.4%           100.0%                   21.4%   78.6%
              1.9%          2.3%   1.7%   2.0%             1.4%                    2.0%    1.6%

0243          186      -     56     56     71      3       4.8       -       -      1.0     3.8    39.0      -     $95.33        -
            100.0%         30.1%  30.1%  38.2%   1.6%    100.0%                   20.3%   79.7%
              1.5%          1.8%   1.2%   1.6%   1.1%      1.0%                    1.3%    1.1%

0260          733      2    184    260    261     26      30.2     2.2       -      4.9    23.1    24.3     0.1   $604.00 $302.00
            100.0%   0.3%  25.1%  35.5%  35.6%   3.5%    100.0%    7.2%           16.3%   76.4%
              5.7%   2.2%   5.8%   5.4%   6.0%   9.5%      6.1%    3.1%            6.7%    6.6%

0267          378      4     65    170    137      2      14.6     3.6       -      1.4     9.6    25.8     0.3   $292.67  $73.17
            100.0%   1.1%  17.2%  45.0%  36.2%   0.5%    100.0%   24.7%            9.5%   65.8%
              3.0%   4.4%   2.0%   3.5%   3.1%   0.7%      3.0%    5.2%            1.9%    2.8%

0268           42      -     16     18      8      -       1.3       -       -      0.4     0.9    33.2      -     $25.33        -
            100.0%         38.1%  42.9%  19.0%           100.0%                   28.9%   71.1%
              0.3%          0.5%   0.4%   0.2%             0.3%                    0.5%    0.3%

0272          262      1     54    112     85     10       9.6       -       -      1.2     8.4    27.2     0.1   $192.33 $192.33
            100.0%   0.4%  20.6%  42.7%  32.4%   3.8%    100.0%                   12.3%   87.7%
              2.1%   1.1%   1.7%   2.3%   1.9%   3.6%      2.0%                    1.6%    2.4%

0282          339      -    120    110    109      -       7.9       -       -      2.1     5.7    43.1      -    $157.33        -
            100.0%         35.4%  32.4%  32.2%           100.0%                   27.1%   72.9%
              2.7%          3.8%   2.3%   2.5%             1.6%                    2.9%    1.6%

0284          512      3    146    184    178      1      17.5     3.3       -      3.1    11.2    29.2     0.2   $350.33 $116.78
            100.0%   0.6%  28.5%  35.9%  34.8%   0.2%    100.0%   18.6%           17.5%   63.8%
              4.0%   3.3%   4.6%   3.8%   4.1%   0.4%      3.6%    4.7%            4.1%    3.2%

0285         1174     14    363    361    405     31      48.7    12.0       -      8.0    28.8    24.1     0.3   $974.67  $69.62
            100.0%   1.2%  30.9%  30.7%  34.5%   2.6%    100.0%   24.5%           16.4%   59.1%
              9.2%  15.4%  11.4%   7.5%   9.3%  11.3%      9.9%   17.1%           10.8%    8.3%

0295           19      -      3      5     10      1       0.9       -       -      0.1     0.7    22.4      -     $17.00        -
            100.0%         15.8%  26.3%  52.6%   5.3%    100.0%                   15.7%   84.3%
              0.1%          0.1%   0.1%   0.2%   0.4%      0.2%                    0.2%    0.2%
```

```
(continued)
                                                    REPORT 3:  INTERVIEWER PRODUCTIVITY REPORT

                       C A L L S    M A D E                     H O U R S    L O G G E D                     S U M M A R Y
             <--------------------------------------------> <------------------------------------------> <---------------------------------->
INTER-        TOTAL   COMP-   OTHER  CALL-   BUSY/  OTHER   TOTAL    COMP-  SUSPEND   OTHER    BUSY,  CALLS    COMP-      TOTAL     COST
VIEWER        CALLS   LETED   RESOL  BACKS     NO  ACTIVE   PHONE    LETED  ED INT-   RESOL     N/A,   PER     LETES    INTRVWER  /COMP-
 ID           MADE   INTVWS    -VED   SCHLD ANSWER STATUS   HOURS  INTRVWS ERVIEWS     -VED    OTHER   HOUR    /HOUR      COST     LETE
------       ------- ------  -----  ----- ------ ------  ------- ------- ------- ------- -------  ------  -----   --------  ------

0316            724      2     182    265    266      9     21.4     2.2       -      3.7     15.5    33.9      0.1    $427.00 $213.50
             100.0%   0.3%   25.1%  36.6%  36.7%   1.2%   100.0%   10.1%            17.3%    72.7%
               5.7%   2.2%    5.7%   5.5%   6.1%   3.3%     4.3%    3.1%             5.0%     4.5%

0320            104      -      23     52     27      2      3.4       -       -      0.7      2.8    30.4        -     $68.33       -
             100.0%          22.1%  50.0%  26.0%   1.9%   100.0%                    19.5%    80.5%
               0.8%           0.7%   1.1%   0.6%   0.7%     0.7%                     0.9%     0.8%

0321            297      -      82     90    123      2      9.3       -       -      1.5      7.8    31.9        -    $186.33       -
             100.0%          27.6%  30.3%  41.4%   0.7%   100.0%                    16.5%    83.5%
               2.3%           2.6%   1.9%   2.8%   0.7%     1.9%                     2.1%     2.2%

0322            465      7      85    160    193     20     19.9     4.0       -      1.6     14.3    23.4      0.4    $397.00  $56.71
             100.0%   1.5%   18.3%  34.4%  41.5%   4.3%   100.0%   20.3%             7.8%    71.9%
               3.6%   7.7%    2.7%   3.3%   4.4%   7.3%     4.0%    5.8%             2.1%     4.1%

0334             21      -       5      9      6      1      0.6       -       -      0.2      0.4    37.1        -     $11.33       -
             100.0%          23.8%  42.9%  28.6%   4.8%   100.0%                    26.5%    73.5%
               0.2%           0.2%   0.2%   0.1%   0.4%     0.1%                     0.2%     0.1%

0342            686      8     138    260    256     24     30.9     8.6       -      2.8     19.5    22.2      0.3    $617.67  $77.21
             100.0%   1.2%   20.1%  37.9%  37.3%   3.5%   100.0%   27.7%             9.1%    63.2%
               5.4%   8.8%    4.3%   5.4%   5.9%   8.7%     6.3%   12.3%             3.8%     5.6%

0343           1070      -     383    295    382     10     40.0       -       -     10.9     29.1    26.8        -    $799.67       -
             100.0%          35.8%  27.6%  35.7%   0.9%   100.0%                    27.1%    72.9%
               8.4%          12.0%   6.1%   8.7%   3.6%     8.1%                    14.6%     8.4%

0344           1362     14     300    561    449     38     54.2     9.7       -      7.6     36.9    25.1      0.3  $1,083.67  $77.40
             100.0%   1.0%   22.0%  41.2%  33.0%   2.8%   100.0%   17.8%            14.0%    68.2%
              10.7%  15.4%    9.4%  11.6%  10.3%  13.8%    11.0%   13.8%            10.2%    10.6%

0349             16      -       2     11      3      -      1.0       -       -      0.1      1.0    15.5        -     $20.67       -
             100.0%          12.5%  68.8%  18.8%          100.0%                     8.1%    91.9%
               0.1%           0.1%   0.2%   0.1%            0.2%                     0.1%     0.3%

0350            145      -      13     69     56      7      7.9       -       -      0.5      7.3    18.5        -    $157.00       -
             100.0%           9.0%  47.6%  38.6%   4.8%   100.0%                     6.6%    93.4%
               1.1%           0.4%   1.4%   1.3%   2.5%     1.6%                     0.7%     2.1%

0351            217      -      45    102     69      1     10.8       -       -      2.0      8.8    20.1        -    $216.33       -
             100.0%          20.7%  47.0%  31.8%   0.5%   100.0%                    18.3%    81.7%
               1.7%           1.4%   2.1%   1.6%   0.4%     2.2%                     2.7%     2.5%

0353            283      -      61     92    126      4     10.5       -       -      1.6      8.9    27.1        -    $209.00       -
             100.0%          21.6%  32.5%  44.5%   1.4%   100.0%                    15.0%    85.0%
               2.2%           1.9%   1.9%   2.9%   1.5%     2.1%                     2.1%     2.5%
```

```
(continued)
                                         REPORT 3:  INTERVIEWER PRODUCTIVITY REPORT

                       C A L L S   M A D E                 H O U R S   L O G G E D                      S U M M A R Y
               <------------------------------------------> <---------------------------------------> <--------------------------------->
INTER-         TOTAL   COMP-  OTHER  CALL-   BUSY/  OTHER   TOTAL   COMP-   SUSPEND  OTHER   BUSY,   CALLS    COMP-     TOTAL    COST
VIEWER         CALLS   LETED  RESOL  BACKS     NO  ACTIVE   PHONE   LETED   ED INT-  RESOL    N/A,    PER     LETES   INTRVWER  /COMP-
 ID            MADE   INTVWS   -VED  SCHLD ANSWER STATUS   HOURS  INTRVWS ERVIEWS   -VED   OTHER   HOUR    /HOUR     COST     LETE
------        ------- ------  -----  ----- ------ ------  ------- ------- ------- ------- -------  ------   -----   --------  ------

INTV              3      1      -      -      -       2      0.0      -       -       -      0.0    90.0     30.0      $0.67   $0.67
             100.0%  33.3%                        66.7%   100.0%                          100.0%
                 *    1.1%                         0.7%       *                                *
```

Note: Percentage less than 0.05 printed as *.

```
                        Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 14

                                 REPORT 4A: CURRENT STATUS OF TELEPHONE NUMBERS BY TIME ZONE (ACTIVE)

                                                         TIMED  CALL- <--------- USER DEFINED CODES ----------><SPEC INTVWR><SYSTEM>
             TOTAL ACTIVE  NOT      NO           BUSY   CALL-  BACK  <--- NO ANSWERS ---><--- TIMED CALLS --->  *1*   *2-9*  (180-
            PHONE  PHONE   YET   ANSWER  BUSY  ->N/A   BACK  UNSPEC               (108-                (162-         (192-   189,
TIME ZONE  NUMBERS NUMBRS CALLED  (101)  (102)  (103)  (104)  (105)  (106)  (107)   159)  (160)  (161)  179)   (191) 199)    200+)
---------  ------- ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------

TOTAL       28130  24823  20958    868     22    343    103    185      -    680      -    817    348    324      -    175      -
            100.0%  88.2%  74.5%   3.1%   0.1%   1.2%   0.4%   0.7%          2.4%          2.9%   1.2%   1.2%          0.6%
            100.0% 100.0% 100.0% 100.0% 100.0% 100.0% 100.0% 100.0%        100.0%        100.0% 100.0% 100.0%        100.0%

EASTERN: 5  15142  13566  11601    423     15    243     45     87      -    350      -    374    168    162      -     98      -
            100.0%  89.6%  76.6%   2.8%   0.1%   1.6%   0.3%   0.6%          2.3%          2.5%   1.1%   1.1%          0.6%
             53.8%  54.7%  55.4%  48.7%  68.2%  70.8%  43.7%  47.0%         51.5%         45.8%  48.3%  50.0%         56.0%

CENTRAL: 6   9866   8688   7325    351      2     90     37     73      -    251      -    272    112    127      -     48      -
            100.0%  88.1%  74.2%   3.6%      *    0.9%   0.4%   0.7%          2.5%          2.8%   1.1%   1.3%          0.5%
             35.1%  35.0%  35.0%  40.4%   9.1%  26.2%  35.9%  39.5%         36.9%         33.3%  32.2%  39.2%         27.4%

MOUNTAIN:7    631    530    418     19      -      2      5      2      -     20      -     28     17     12      -      7      -
            100.0%  84.0%  66.2%   3.0%          0.3%   0.8%   0.3%          3.2%          4.4%   2.7%   1.9%          1.1%
              2.2%   2.1%   2.0%   2.2%          0.6%   4.9%   1.1%          2.9%          3.4%   4.9%   3.7%          4.0%

PACIFIC: 8   2491   2039   1614     75      5      8     16     23      -     59      -    143     51     23      -     22      -
            100.0%  81.9%  64.8%   3.0%   0.2%   0.3%   0.6%   0.9%          2.4%          5.7%   2.0%   0.9%          0.9%
              8.9%   8.2%   7.7%   8.6%  22.7%   2.3%  15.5%  12.4%          8.7%         17.5%  14.7%   7.1%         12.6%
```

Note: Percentage less than 0.05 printed as *.

```
                        Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 15

                                  REPORT 4B: CURRENT STATUS OF TELEPHONE NUMBERS BY TIME ZONE (RESOLVED)

                                  LANG-   TERMI  NON-  NON-    NON-  <---------  TERMINATES  ---------> USER   AUTO  OTHER   MAX.
             TOTAL  COMP-   RE-  UAGE   -NATED WORKNG RESI-   BUSI-  TERM.  TERM.  TERM.  TERM.  TERM. CODES -DIAL SYSTEM  CALLS
           RESOLVD  LETES  FUSED PROBLM INTRVW NUMBER DENTAL  NESS     A      B      C      D      E    (13-  (70- (80-90,  MADE
TIME ZONE   (1-99)   (1)    (2)    (3)    (4)    (5)    (6)    (7)    (8)    (9)   (10)   (11)   (12)    69) 79,82)  92+    (94)
---------  ------- ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------

TOTAL        3307     86   1191     14      5   1354      -      -    126     61      -      -      -    266    156      -     48
            100.0%   2.6%  36.0%   0.4%   0.2%  40.9%                3.8%   1.8%                        8.0%   4.7%          1.5%
            100.0% 100.0% 100.0% 100.0% 100.0% 100.0%              100.0% 100.0%                      100.0% 100.0%        100.0%

EASTERN: 5   1576     39    570      9      2    637      -      -     58     34      -      -      -    147     65      -     15
            100.0%   2.5%  36.2%   0.6%   0.1%  40.4%                3.7%   2.2%                        9.3%   4.1%          1.0%
             47.7%  45.3%  47.9%  64.3%  40.0%  47.0%               46.0%  55.7%                       55.3%  41.7%         31.3%

CENTRAL: 6   1178     26    455      2      1    478      -      -     45     14      -      -      -     79     68      -     10
            100.0%   2.2%  38.6%   0.2%   0.1%  40.6%                3.8%   1.2%                        6.7%   5.8%          0.8%
             35.6%  30.2%  38.2%  14.3%  20.0%  35.3%               35.7%  23.0%                       29.7%  43.6%         20.8%

MOUNTAIN:7    101      1     45      -      2     34      -      -      5      3      -      -      -      5      3      -      3
            100.0%   1.0%  44.6%          2.0%  33.7%                5.0%   3.0%                        5.0%   3.0%          3.0%
              3.1%   1.2%   3.8%         40.0%   2.5%                4.0%   4.9%                        1.9%   1.9%          6.3%

PACIFIC: 8    452     20    121      3      -    205      -      -     18     10      -      -      -     35     20      -     20
            100.0%   4.4%  26.8%   0.7%         45.4%                4.0%   2.2%                        7.7%   4.4%          4.4%
             13.7%  23.3%  10.2%  21.4%         15.1%               14.3%  16.4%                       13.2%  12.8%         41.7%
```

```
                        Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 16

                               REPORT 5A:  NUMBER OF CALLS MADE TO ACTIVE PHONE NUMBERS BY LAST STATUS

                                                <--------------------------  NUMBER OF CALLS MADE  ------------------------>               AVERAGE
                               TOTAL                                                                                 10 OR       TOTAL   # CALLS
STATUS OF LAST CALL           ACTIVE   NONE     ONE     TWO    THREE  FOUR   FIVE     SIX    SEVEN  EIGHT  NINE   MORE        CALLS     MADE
-------------------          -------- ------  -----   -----    ----- ------ ------   -----   -----  ----- ------ ------      ------- -------

TOTAL                          24823  20958   1493    1357     697    227     69      16      4      1      -      1         7693      0.31
                              100.0% 100.0% 100.0% 100.0% 100.0% 100.0% 100.0% 100.0% 100.0% 100.0%        100.0%       100.0%

  0  - FRESH NUMBERS          20958  20958     -       -       -      -      -       -      -      -      -      -            -      0.00
                              84.4% 100.0%

101 - NO ANSWER                 868      -    528     227     101     11      1       -      -      -      -      -         1334      1.54
                                3.5%         35.4%   16.7%   14.5%   4.8%   1.4%                                            17.3%

102 - BUSY (TIMED CALL)          22      -      8       4       9      -      1       -      -      -      -      -           48      2.18
                                0.1%          0.5%    0.3%    1.3%          1.4%                                             0.6%

103 - BUSY-->NO ANSWER          343      -      -     232      72     33      4       2      -      -      -      -          844      2.46
                                1.4%                 17.1%   10.3%  14.5%   5.8%   12.5%                                    11.0%

104 - TIMED CALL-BACK           103      -     29      27      26     10      8       1      1      -      -      1          264      2.56
                                0.4%          1.9%    2.0%    3.7%   4.4%  11.6%    6.3%  25.0%               100.0%         3.4%

105 - TIMED CALL-->NO ANSWER    185      -    107      44      22      9      3       -      -      -      -      -          312      1.69
                                0.7%          7.2%    3.2%    3.2%   4.0%   4.3%                                             4.1%

USER NO ANSWERS (106-159):      680      -    425     169      59     20      7       -      -      -      -      -         1055      1.55
                                2.7%         28.5%   12.5%    8.5%   8.8%  10.1%                                            13.7%
  STATUS 107                    680      -    425     169      59     20      7       -      -      -      -      -         1055      1.55
                                2.7%         28.5%   12.5%    8.5%   8.8%  10.1%                                            13.7%

USER TIMED CALLS (160-179):    1489      -    372     596     346    124     37      11      2      1      -      -         3371      2.26
                                6.0%         24.9%   43.9%   49.6%  54.6%  53.6%   68.8%  50.0% 100.0%                      43.8%
  STATUS 160                    817      -    214     312     177     81     27       5      1      -      -      -         1865      2.28
                                3.3%         14.3%   23.0%   25.4%  35.7%  39.1%   31.3%  25.0%                             24.2%
  STATUS 161                    348      -     54     148     112     24      5       3      1      1      -      -          840      2.41
                                1.4%          3.6%   10.9%   16.1%  10.6%   7.2%   18.8%  25.0% 100.0%                      10.9%
  STATUS 162                    324      -    104     136      57     19      5       3      -      -      -      -          666      2.06
                                1.3%          7.0%   10.0%    8.2%   8.4%   7.2%   18.8%                                     8.7%

SYSTEM STATUSES (180-214):      175      -     24      58      62     20      8       2      1      -      -      -          465      2.66
                                0.7%          1.6%    4.3%    8.9%   8.8%  11.6%   12.5%  25.0%                              6.0%
198 - SPECIAL INTERVIEWER 8      18      -      5       6       3      2      2       -      -      -      -      -           44      2.44
                                0.1%          0.3%    0.4%    0.4%   0.9%   2.9%                                             0.6%
199 - SPECIAL INTERVIEWER 9     157      -     19      52      59     18      6       2      1      -      -      -          421      2.68
                                0.6%          1.3%    3.8%    8.5%   7.9%   8.7%   12.5%  25.0%                              5.5%
```

Computers for Marketing - Phone report for eid572.fon - 1:23 PM 13 FEB 2001 - Page 17

REPORT 5B: NUMBER OF CALLS MADE TO RESOLVED PHONE NUMBERS BY FINAL STATUS

		<------------------------- NUMBER OF CALLS MADE ----------------------->											AVERAGE
STATUS OF FINAL CALL	TOTAL RESOLVED	NONE	ONE	TWO	THREE	FOUR	FIVE	SIX	SEVEN	EIGHT	NINE	10 OR MORE	TOTAL CALLS	# CALLS MADE
TOTAL	3307	-	2160	743	241	96	40	19	6	2	-	-	5125	1.55
	100.0%		100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%			100.0%	
1 - COMPLETED INTERVIEW	86	-	1	14	39	21	9	2	-	-	-	-	287	3.34
	2.6%		*	1.9%	16.2%	21.9%	22.5%	10.5%					5.6%	
2 - REFUSED TO START	1191	-	641	384	121	29	10	4	1	1	-	-	1977	1.66
	36.0%		29.7%	51.7%	50.2%	30.2%	25.0%	21.1%	16.7%	50.0%			38.6%	
3 - LANGUAGE PROBLEM	14	-	7	6	-	1	-	-	-	-	-	-	23	1.64
	0.4%		0.3%	0.8%		1.0%							0.4%	
4 - TERMINATED INTERVIEW	5	-	1	3	1	-	-	-	-	-	-	-	10	2.00
	0.2%		*	0.4%	0.4%								0.2%	
5 - NON-WORKING NUMBER	1354	-	1164	154	27	5	-	2	2	-	-	-	1599	1.18
	40.9%		53.9%	20.7%	11.2%	5.2%		10.5%	33.3%				31.2%	
8 - TERMINATE REASON A	126	-	97	24	3	1	1	-	-	-	-	-	163	1.29
	3.8%		4.5%	3.2%	1.2%	1.0%	2.5%						3.2%	
9 - TERMINATE REASON B	61	-	46	12	2	1	-	-	-	-	-	-	80	1.31
	1.8%		2.1%	1.6%	0.8%	1.0%							1.6%	
NET USER DEFINED (13-70):	266	-	81	116	43	20	3	2	-	1	-	-	557	2.09
	8.0%		3.8%	15.6%	17.8%	20.8%	7.5%	10.5%		50.0%			10.9%	
STATUS 22	21	-	4	7	7	3	-	-	-	-	-	-	51	2.43
	0.6%		0.2%	0.9%	2.9%	3.1%							1.0%	
STATUS 23	2	-	-	1	1	-	-	-	-	-	-	-	5	2.50
	0.1%			0.1%	0.4%								0.1%	
STATUS 24	6	-	1	3	1	-	-	-	-	1	-	-	18	3.00
	0.2%		*	0.4%	0.4%					50.0%			0.4%	
STATUS 25	2	-	-	-	2	-	-	-	-	-	-	-	6	3.00
	0.1%				0.8%								0.1%	
STATUS 27	4	-	-	3	-	1	-	-	-	-	-	-	10	2.50
	0.1%			0.4%		1.0%							0.2%	
STATUS 28	3	-	-	-	2	1	-	-	-	-	-	-	10	3.33
	0.1%				0.8%	1.0%							0.2%	
STATUS 67	9	-	5	3	1	-	-	-	-	-	-	-	14	1.56
	0.3%		0.2%	0.4%	0.4%								0.3%	
STATUS 68	219	-	71	99	29	15	3	2	-	-	-	-	443	2.02
	6.6%		3.3%	13.3%	12.0%	15.6%	7.5%	10.5%					8.6%	
STATUS 70	156	-	121	29	2	3	1	-	-	-	-	-	202	1.29
	4.7%		5.6%	3.9%	0.8%	3.1%	2.5%						3.9%	
NET SYSTEM NUMBERS (80-100):	48	-	1	1	3	15	16	9	3	-	-	-	227	4.73
	1.5%		*	0.1%	1.2%	15.6%	40.0%	47.4%	50.0%				4.4%	
94 - MAXIMUM # OF CALLS	48	-	1	1	3	15	16	9	3	-	-	-	227	4.73
	1.5%		*	0.1%	1.2%	15.6%	40.0%	47.4%	50.0%				4.4%	

Note: Percentage less than 0.05 printed as *.

```
                         Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 18

                                     REPORT 6:  TIME ZONE BY TELEPHONE NUMBER AVAILABILITY (STACK)

                                            <-------------------------------------- TIME ZONE ------------------------------------>
                                 TOTAL      (5)       (6)       (7)        (8)       (9)       (10)      (23)      (24)     (1-4,11+)
AVAILABILITY (STACK)            NUMBERS   EASTERN   CENTRAL  MOUNTAIN    PACIFIC   ALASKA     HAWAII   E.EUROPE  W.EUROPE   OTHER TZ
--------------------           --------- --------- --------- --------- --------- --------- --------- --------- --------- ---------

TOTAL                             28130     15142      9866       631      2491          -         -         -         -         -
                                 100.0%     53.8%     35.1%      2.2%      8.9%
                                 100.0%    100.0%    100.0%    100.0%    100.0%

TOTAL AVAILABLE (% base):        24823     13566      8688       530      2039           -         -         -         -         -
                                 100.0%     54.7%     35.0%      2.1%      8.2%
                                  88.2%     89.6%     88.1%     84.0%     81.9%

TOTAL SYSTEM NUMBERS:            23031     12702      8089       461      1779           -         -         -         -         -
                                 100.0%     55.2%     35.1%      2.0%      7.7%
                                  92.8%     93.6%     93.1%     87.0%     87.2%

  NEW NUMBERS                     20955     11599      7324       418      1614          -         -         -         -         -
                                 100.0%     55.4%     35.0%      2.0%      7.7%
                                  84.4%     85.5%     84.3%     78.9%     79.2%
  DAY PART 1 ONLY                 113        35        48         -        30           -         -         -         -         -
                                 100.0%     31.0%     42.5%               26.5%
                                   0.5%      0.3%      0.6%                1.5%
  DAY PART 2 ONLY                  67        13        41         7         6           -         -         -         -         -
                                 100.0%     19.4%     61.2%     10.4%      9.0%
                                   0.3%      0.1%      0.5%      1.3%      0.3%
  DAY PART 3 ONLY                  61        42        19         -         -           -         -         -         -         -
                                 100.0%     68.9%     31.1%
                                   0.2%      0.3%      0.2%
  DAY PART 1 OR 2                 592       232       225        28       107           -         -         -         -         -
                                 100.0%     39.2%     38.0%      4.7%     18.1%
                                   2.4%      1.7%      2.6%      5.3%      5.2%
  DAY PART 1 OR 3                 643       410       232         -         1           -         -         -         -         -
                                 100.0%     63.8%     36.1%                0.2%
                                   2.6%      3.0%      2.7%                 *
  DAY PART 2 OR 3                 552       362       190         -         -           -         -         -         -         -
                                 100.0%     65.6%     34.4%
                                   2.2%      2.7%      2.2%
  DAY PART 1, 2, OR 3               3         1         2         -         -           -         -         -         -         -
                                 100.0%     33.3%     66.7%
                                    *         *         *
  ALL TARGETED ATTEMPTS MADE       45         8         8         8        21           -         -         -         -         -
                                 100.0%     17.8%     17.8%     17.8%     46.7%
                                   0.2%      0.1%      0.1%      1.5%      1.0%
  ***NET DAY PART 1 CALLS        1351       678       507        28       138           -         -         -         -         -
                                 100.0%     50.2%     37.5%      2.1%     10.2%
                                   5.4%      5.0%      5.8%      5.3%      6.8%
  ***NET DAY PART 2 CALLS        1214       608       458        35       113           -         -         -         -         -
                                 100.0%     50.1%     37.7%      2.9%      9.3%
                                   4.9%      4.5%      5.3%      6.6%      5.5%
  ***NET DAY PART 3 CALLS        1259       815       443         -         1           -         -         -         -         -
                                 100.0%     64.7%     35.2%                0.1%
                                   5.1%      6.0%      5.1%                 *
```

```
                      Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 19

(continued)
                                    REPORT 6:  TIME ZONE BY TELEPHONE NUMBER AVAILABILITY (STACK)

                                           <-------------------------------------- TIME ZONE ------------------------------------->
                              TOTAL     (5)       (6)       (7)       (8)       (9)       (10)      (23)      (24)    (1-4,11+)
AVAILABILITY (STACK)         NUMBERS   EASTERN   CENTRAL  MOUNTAIN   PACIFIC   ALASKA    HAWAII   E.EUROPE  W.EUROPE   OTHER TZ
--------------------         --------- --------- --------- --------- --------- --------- --------- --------- --------- ---------

  TOTAL TIMED TODAY:            1089      445       382       49       213         -         -         -         -         -
                              100.0%    40.9%     35.1%     4.5%     19.6%
                                4.4%     3.3%      4.4%     9.2%     10.4%
     SPECIFIC CALLBACKS:        1089      445       382       49       213         -         -         -         -         -
                              100.0%    40.9%     35.1%     4.5%     19.6%
                                4.4%     3.3%      4.4%     9.2%     10.4%
  ***MORNING (6am - <12pm)       262      177        70       12         3         -         -         -         -         -
                              100.0%    67.6%     26.7%     4.6%      1.1%
                                1.1%     1.3%      0.8%     2.3%      0.1%
  ***AFTERNOON (12pm - <6pm)     686      267       298       22        99         -         -         -         -         -
                              100.0%    38.9%     43.4%     3.2%     14.4%
                                2.8%     2.0%      3.4%     4.2%      4.9%
  ***EVENING (6pm - <12am)       141        1        14       15       111         -         -         -         -         -
                              100.0%     0.7%      9.9%    10.6%     78.7%
                                0.6%        *      0.2%     2.8%      5.4%

  TIMED CALLBACKS LATER:         527      320       168       13        26         -         -         -         -         -
                              100.0%    60.7%     31.9%     2.5%      4.9%
                                2.1%     2.4%      1.9%     2.5%      1.3%

  SPECIAL INTERVIEWER ONLY       176       99        49        7        21         -         -         -         -         -
                              100.0%    56.3%     27.8%     4.0%     11.9%
                                0.7%     0.7%      0.6%     1.3%      1.0%

RESOLVED NUMBERS: (% base)      3307     1576      1178      101       452         -         -         -         -         -
                              100.0%    47.7%     35.6%     3.1%     13.7%
                               11.8%    10.4%     11.9%    16.0%     18.1%

  COMPLETES                       86       39        26        1        20         -         -         -         -         -
                              100.0%    45.3%     30.2%     1.2%     23.3%
                                   -        -         -        -         -

  NON-COMPLETES                 3173     1522      1142       97       412         -         -         -         -         -
                              100.0%    48.0%     36.0%     3.1%     13.0%
                                   -        -         -        -         -

  MAX CALLS/MAX HISTORIES         48       15        10        3        20         -         -         -         -         -
                              100.0%    31.3%     20.8%     6.3%     41.7%
                                   -        -         -        -         -

  OTHER SYSTEM RESOLVED           48       15        10        3        20         -         -         -         -         -
                              100.0%    31.3%     20.8%     6.3%     41.7%
                                   -        -         -        -         -



Note: Percentage less than 0.05 printed as *.
```

```
                         Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 20

                         REPORT 7:  NUMBER OF CALLS MADE BY ACTIVE NUMBER AVAILABILITY (STACK) WITHIN TIME ZONE

                                        NOT   <---------------------  NUMBER OF CALLS MADE  ---------------------->          AVERAGE
                              TOTAL    YET                                                                10 OR      TOTAL # CALLS
TIME ZONE/AVAILABILITY       ACTIVE  CALLED   ONE    TWO   THREE  FOUR   FIVE    SIX   SEVEN  EIGHT  NINE   MORE      CALLS   MADE
----------------------      -------- ------  -----  -----  ----- ------ ------  -----  -----  ------ ------ ------    ------ -------

**TOTAL AVAILABLE:            24823  20958   1493   1357    697    227     69     16      4      1      -      1       7693    0.31
                             100.0% 100.0% 100.0% 100.0% 100.0% 100.0% 100.0% 100.0% 100.0% 100.0%        100.0%     100.0%

  SYSTEM SCHEDULED:           23031  20955   1060    672    254     73     15      2      -      -      -      -       3545    0.15
                              92.8% 100.0%  71.0%  49.5%  36.4%  32.2%  21.7%  12.5%                                   46.1%
    NEW NUMBERS               20955  20955      -      -      -      -      -      -      -      -      -      -          -    0.00
                              84.4% 100.0%
    DAY PART 1 ONLY             113      -      -     52     50      7      4      -      -      -      -      -        302    2.67
                               0.5%                 3.8%   7.2%   3.1%   5.8%                                           3.9%
    DAY PART 2 ONLY              67      -      -     29     26      9      2      1      -      -      -      -        188    2.81
                               0.3%                 2.1%   3.7%   4.0%   2.9%   6.3%                                    2.4%
    DAY PART 3 ONLY              61      -      -     24     25     11      1      -      -      -      -      -        172    2.82
                               0.2%                 1.8%   3.6%   4.8%   1.4%                                           2.2%
    DAY PART 1 OR 2             592      -    370    176     39      7      -      -      -      -      -      -        867    1.46
                               2.4%         24.8%  13.0%   5.6%   3.1%                                                 11.3%
    DAY PART 1 OR 3             643      -    368    214     53      8      -      -      -      -      -      -        987    1.53
                               2.6%         24.6%  15.8%   7.6%   3.5%                                                 12.8%
    DAY PART 2 OR 3             552      -    319    177     42     13      1      -      -      -      -      -        856    1.55
                               2.2%         21.4%  13.0%   6.0%   5.7%   1.4%                                          11.1%
    DAY PART 1, 2, OR 3           3      -      3      -      -      -      -      -      -      -      -      -          3    1.00
                                  *          0.2%                                                                          *
    DAY PART TARGETS MET         45      -      -      -     19     18      7      1      -      -      -      -        170    3.78
                               0.2%                        2.7%   7.9%  10.1%   6.3%                                    2.2%

  TIMED CALLBACKS TODAY        1616      2    409    628    380    134     46     12      3      1      -      1       3682    2.28
                               6.5%      *  27.4%  46.3%  54.5%  59.0%  66.7%  75.0%  75.0% 100.0%        100.0%      47.9%

  ***TIMED MORNING (6-<12)      262      -     81     98     59     15      7      2      -      -      -      -        561    2.14
                               1.1%          5.4%   7.2%   8.5%   6.6%  10.1%  12.5%                                    7.3%

  ***TIMED AFTERNOON (12-<6)    686      -    124    310    176     53     18      4      1      -      -      -       1605    2.34
                               2.8%          8.3%  22.8%  25.3%  23.3%  26.1%  25.0%  25.0%                            20.9%

  ***TIMED EVENING (6-<12)      141      -     37     56     28     16      3      1      -      -      -      -        318    2.26
                               0.6%          2.5%   4.1%   4.0%   7.0%   4.3%   6.3%                                    4.1%

  TIMED LATER                   527      2    167    164    117     50     18      5      2      1      -      1       1198    2.27
                               2.1%      *  11.2%  12.1%  16.8%  22.0%  26.1%  31.3%  50.0% 100.0%        100.0%      15.6%

  SPECIAL INTERVIEWER ONLY      176      1     24     57     63     20      8      2      1      -      -      -        466    2.65
                               0.7%      *   1.6%   4.2%   9.0%   8.8%  11.6%  12.5%  25.0%                             6.1%

**EASTERN TZ AVAILABLE:       13566  11601    778    701    328    119     30      4      4      1      -      -       3850    0.28
                              54.7%  55.4%  52.1%  51.7%  47.1%  52.4%  43.5%  25.0% 100.0% 100.0%                     50.0%

  SYSTEM SCHEDULED:           12702  11599    569    368    113     47      5      1      -      -      -      -       1863    0.15
                              51.2%  55.3%  38.1%  27.1%  16.2%  20.7%   7.2%   6.3%                                   24.2%
    NEW NUMBERS               11599  11599      -      -      -      -      -      -      -      -      -      -          -    0.00
                              46.7%  55.3%
```

```
                         Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 21

(continued)
                         REPORT 7:  NUMBER OF CALLS MADE BY ACTIVE NUMBER AVAILABILITY (STACK) WITHIN TIME ZONE

                                        NOT   <---------------------  NUMBER OF CALLS MADE  ---------------------->        AVERAGE
                               TOTAL    YET                                                             10 OR      TOTAL # CALLS
TIME ZONE/AVAILABILITY        ACTIVE  CALLED   ONE    TWO   THREE  FOUR   FIVE    SIX   SEVEN  EIGHT  NINE   MORE      CALLS   MADE
---------------------        -------- ------  -----  -----  ----- ------ ------  -----  -----  ------ ------ ------    ------ -------

    DAY PART 1 ONLY             35      -      -      13     16     4      2      -      -      -      -      -         100    2.86
                               0.1%                  1.0%   2.3%   1.8%   2.9%                                         1.3%
    DAY PART 2 ONLY             13      -      -       6      4     3      -      -      -      -      -      -          36    2.77
                               0.1%                  0.4%   0.6%   1.3%                                                0.5%
    DAY PART 3 ONLY             42      -      -      16     14    11      1      -      -      -      -      -         123    2.93
                               0.2%                  1.2%   2.0%   4.8%   1.4%                                         1.6%
    DAY PART 1 OR 2            232      -    127      79     19     7      -      -      -      -      -      -         370    1.59
                               0.9%         8.5%     5.8%   2.7%   3.1%                                                4.8%
    DAY PART 1 OR 3            410      -    232     142     31     5      -      -      -      -      -      -         629    1.53
                               1.7%        15.5%    10.5%   4.4%   2.2%                                                8.2%
    DAY PART 2 OR 3            362      -    209     112     28    12      1      -      -      -      -      -         570    1.57
                               1.5%        14.0%     8.3%   4.0%   5.3%   1.4%                                         7.4%
    DAY PART 1, 2 OR 3           1      -      1       -      -     -      -      -      -      -      -      -           1    1.00
                                *           0.1%                                                                         *
    DAY PART TARGETS MET         8      -      -       -      1     5      1      1      -      -      -      -          34    4.25
                                *                         0.1%   2.2%   1.4%   6.3%                                  0.4%

  TIMED CALLBACKS TODAY        765      2    193     300    180    63     21      2      3      1      -      -        1731    2.26
                               3.1%     *  12.9%   22.1%  25.8%  27.8%  30.4%  12.5%  75.0% 100.0%                     22.5%

  ***TIMED MORNING (6-<12)     177      -     55      65     41    12      4      -      -      -      -      -         376    2.12
                               0.7%         3.7%     4.8%   5.9%   5.3%   5.8%                                         4.9%

  ***TIMED AFTERNOON (12-<6)   267      -     41     141     66    16      2      -      1      -      -      -         602    2.25
                               1.1%         2.7%    10.4%   9.5%   7.0%   2.9%         25.0%                           7.8%

  ***TIMED EVENING (6-<12)       1      -      -       -      1     -      -      -      -      -      -      -           3    3.00
                                *                         0.1%                                                           *

  TIMED LATER                  320      2     97      94     72    35     15      2      2      1      -      -         750    2.34
                               1.3%     *   6.5%     6.9%  10.3%  15.4%  21.7%  12.5%  50.0% 100.0%                      9.7%

  SPECIAL INTERVIEWER ONLY      99      -     16      33     35     9      4      1      1      -      -      -         256    2.59
                               0.4%         1.1%     2.4%   5.0%   4.0%   5.8%   6.3%  25.0%                           3.3%

**CENTRAL TZ AVAILABLE:       8688   7325    590     483    221    50     14      4      -      -      -      1        2523    0.29
                              35.0%  35.0%  39.5%   35.6%  31.7%  22.0%  20.3%  25.0%                      100.0%     32.8%

  SYSTEM SCHEDULED:           8089   7324    411     245     91    15      3      -      -      -      -      -        1249    0.15
                              32.6%  34.9%  27.5%   18.1%  13.1%   6.6%   4.3%                                        16.2%
    NEW NUMBERS               7324   7324      -       -      -     -      -      -      -      -      -      -           -    0.00
                              29.5%  34.9%
    DAY PART 1 ONLY             48      -      -      28     19     1      -      -      -      -      -      -         117    2.44
                               0.2%                  2.1%   2.7%   0.4%                                                1.5%
    DAY PART 2 ONLY             41      -      -      18     17     6      -      -      -      -      -      -         111    2.71
                               0.2%                  1.3%   2.4%   2.6%                                                1.4%
    DAY PART 3 ONLY             19      -      -       8     11     -      -      -      -      -      -      -          49    2.58
                               0.1%                  0.6%   1.6%                                                       0.6%
                         Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 22

(continued)
                         REPORT 7:  NUMBER OF CALLS MADE BY ACTIVE NUMBER AVAILABILITY (STACK) WITHIN TIME ZONE

                                        NOT    <---------------------  NUMBER OF CALLS MADE  ---------------------->          AVERAGE
                               TOTAL   YET                                                               10 OR     TOTAL # CALLS
TIME ZONE/AVAILABILITY       ACTIVE  CALLED   ONE    TWO   THREE  FOUR   FIVE     SIX   SEVEN  EIGHT  NINE   MORE      CALLS   MADE
----------------------       -------- ------   -----  -----  ----- ------ ------   -----  -----  ------ ------ ------   ------  -------

    DAY PART 1 OR 2             225       -    163     54      8       -      -      -      -      -       -      -       295    1.31
                                 0.9%         10.9%   4.0%   1.1%                                                         3.8%
    DAY PART 1 OR 3             232       -    136     72     21       3      -      -      -      -       -      -       355    1.53
                                 0.9%          9.1%   5.3%   3.0%    1.3%                                                 4.6%
    DAY PART 2 OR 3             190       -    110     65     14       1      -      -      -      -       -      -       286    1.51
                                 0.8%          7.4%   4.8%   2.0%    0.4%                                                 3.7%
    DAY PART 1, 2, OR 3           2       -      2      -      -       -      -      -      -      -       -      -         2    1.00
                                   *           0.1%                                                                         *
    DAY PART TARGETS MET          8       -      -      -      1       4      3      -      -      -       -      -        34    4.25
                                   *                         0.1%    1.8%   4.3%                                          0.4%

  TIMED CALLBACKS TODAY         550       -    173    221    113     29       9      4      -      -       -      1      1149    2.09
                                 2.2%         11.6%  16.3%  16.2%   12.8%  13.0%  25.0%                       100.0%     14.9%

  ***TIMED MORNING (6-<12)       70       -     24     27     14       2      2      1      -      -       -      -       144    2.06
                                 0.3%          1.6%   2.0%   2.0%    0.9%   2.9%   6.3%                                   1.9%

  ***TIMED AFTERNOON (12-<6)    298       -     78    125     72     17       5      1      -      -       -      -       643    2.16
                                 1.2%          5.2%   9.2%  10.3%    7.5%   7.2%   6.3%                                   8.4%

  ***TIMED EVENING (6-<12)       14       -      1      9      1       2      1      -      -      -       -      -        35    2.50
                                 0.1%          0.1%   0.7%   0.1%    0.9%   1.4%                                          0.5%

  TIMED LATER                   168       -     70     60     26       8      1      2      -      -       -      1       327    1.95
                                 0.7%          4.7%   4.4%   3.7%    3.5%   1.4%  12.5%                       100.0%      4.3%

  SPECIAL INTERVIEWER ONLY       49       1      6     17     17       6      2      -      -      -       -      -       125    2.55
                                 0.2%     *    0.4%   1.3%   2.4%    2.6%   2.9%                                          1.6%

**MOUNTAIN TZ AVAILABLE:        530    418     25     34     34     10       4      5      -      -       -      -       285    0.54
                                 2.1%   2.0%   1.7%   2.5%   4.9%    4.4%   5.8%  31.3%                                   3.7%

  SYSTEM SCHEDULED:             461    418     14     11     15       2      -      1      -      -       -      -        95    0.21
                                 1.9%   2.0%   0.9%   0.8%   2.2%    0.9%          6.3%                                   1.2%
    NEW NUMBERS                 418    418      -      -      -       -      -      -      -      -       -      -         -    0.00
                                 1.7%   2.0%
    DAY PART 2 ONLY               7       -      -      3      3       -      -      1      -      -       -      -        21    3.00
                                   *                  0.2%   0.4%                  6.3%                                   0.3%
    DAY PART 1 OR 2              28       -     14      8      6       -      -      -      -      -       -      -        48    1.71
                                 0.1%          0.9%   0.6%   0.9%                                                         0.6%
    DAY PART TARGETS MET          8       -      -      -      6       2      -      -      -      -       -      -        26    3.25
                                   *                         0.9%    0.9%                                                 0.3%

  TIMED CALLBACKS TODAY          62       -     10     22     15       8      4      3      -      -       -      -       169    2.73
                                 0.2%          0.7%   1.6%   2.2%    3.5%   5.8%  18.8%                                   2.2%

  ***TIMED MORNING (6-<12)       12       -      2      4      4       1      1      -      -      -       -      -        31    2.58
                                   *           0.1%   0.3%   0.6%    0.4%   1.4%                                          0.4%
```

```
                          Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 23

(continued)
                          REPORT 7:  NUMBER OF CALLS MADE BY ACTIVE NUMBER AVAILABILITY (STACK) WITHIN TIME ZONE

                                       NOT   <---------------------  NUMBER OF CALLS MADE  ---------------------->         AVERAGE
                              TOTAL   YET                                                              10 OR     TOTAL # CALLS
TIME ZONE/AVAILABILITY       ACTIVE  CALLED   ONE    TWO   THREE  FOUR   FIVE    SIX   SEVEN  EIGHT  NINE   MORE      CALLS   MADE
----------------------      -------- ------  -----  -----  ----- ------ ------  -----  ----- ------ ------ ------    ------ -------

  ***TIMED AFTERNOON (12-<6)     22     -       5      9      1      3      2      2      -      -      -      -        60    2.73
                                0.1%          0.3%   0.7%   0.1%   1.3%   2.9%  12.5%                                0.8%

  ***TIMED EVENING (6-<12)       15     -       3      5      4      2      -      1      -      -      -      -        39    2.60
                                0.1%          0.2%   0.4%   0.6%   0.9%          6.3%                                0.5%

  TIMED LATER                    13     -       -      4      6      2      1      -      -      -      -      -        39    3.00
                                0.1%                 0.3%   0.9%   0.9%   1.4%                                       0.5%

  SPECIAL INTERVIEWER ONLY        7     -       1      1      4      -      -      1      -      -      -      -        21    3.00
                                  *           0.1%   0.1%   0.6%                 6.3%                                0.3%

**PACIFIC TZ AVAILABLE:        2039   1614    100    139    114     48     21      3      -      -      -      -      1035    0.51
                                8.2%   7.7%   6.7%  10.2%  16.4%  21.1%  30.4%  18.8%                               13.5%

  SYSTEM SCHEDULED:            1779   1614     66     48     35      9      7      -      -      -      -      -       338    0.19
                                7.2%   7.7%   4.4%   3.5%   5.0%   4.0%  10.1%                                       4.4%
    NEW NUMBERS                1614   1614      -      -      -      -      -      -      -      -      -      -         -    0.00
                                6.5%   7.7%
    DAY PART 1 ONLY              30     -       -     11     15      2      2      -      -      -      -      -        85    2.83
                                0.1%                 0.8%   2.2%   0.9%   2.9%                                       1.1%
    DAY PART 2 ONLY               6     -       -      2      2      -      2      -      -      -      -      -        20    3.33
                                  *                  0.1%   0.3%          2.9%                                       0.3%
    DAY PART 1 OR 2             107     -      66     35      6      -      -      -      -      -      -      -       154    1.44
                                0.4%          4.4%   2.6%   0.9%                                                     2.0%
    DAY PART 1 OR 3               1     -       -      -      1      -      -      -      -      -      -      -         3    3.00
                                  *                         0.1%                                                       *
    DAY PART TARGETS MET         21     -       -      -     11      7      3      -      -      -      -      -        76    3.62
                                0.1%                        1.6%   3.1%   4.3%                                       1.0%

  TIMED CALLBACKS TODAY         239     -      33     85     72     34     12      3      -      -      -      -       633    2.65
                                1.0%          2.2%   6.3%  10.3%  15.0%  17.4%  18.8%                                8.2%

  ***TIMED MORNING (6-<12)        3     -       -      2      -      -      -      1      -      -      -      -        10    3.33
                                  *                  0.1%                        6.3%                                0.1%

  ***TIMED AFTERNOON (12-<6)     99     -       -     35     37     17      9      1      -      -      -      -       300    3.03
                                0.4%                 2.6%   5.3%   7.5%  13.0%   6.3%                                3.9%

  ***TIMED EVENING (6-<12)      111     -      33     42     22     12      2      -      -      -      -      -       241    2.17
                                0.4%          2.2%   3.1%   3.2%   5.3%   2.9%                                       3.1%

  TIMED LATER                    26     -       -      6     13      5      1      1      -      -      -      -        82    3.15
                                0.1%                 0.4%   1.9%   2.2%   1.4%   6.3%                                1.1%

  SPECIAL INTERVIEWER ONLY       21     -       1      6      7      5      2      -      -      -      -      -        64    3.05
                                0.1%          0.1%   0.4%   1.0%   2.2%   2.9%                                       0.8%

Note: Percentage less than 0.05 printed as *.
```

```
                         Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 24

                        REPORT 8:  SPECIAL INTERVIEW TYPE BY ACTIVE NUMBER AVAILABILITY (STACK) WITHIN TIME ZONE

                                        NOT   <================ SPECIAL INTERVIEWER TYPES =================>
                               TOTAL SPECIAL   TYPE   TYPE   TYPE   TYPE   TYPE   TYPE   TYPE   TYPE   TYPE
TIME ZONE/AVAILABILITY        ACTIVE  TYPE      1      2      3      4      5      6      7      8      9
----------------------        ------ ------ ---- ---- ---- ---- ---- ---- ----   ----   ----   ----   ----

**TOTAL AVAILABLE:             24823  24642     -      -      -      -      -      -      -     22    159
                              100.0% 100.0%                                                100.0% 100.0%

  SYSTEM SCHEDULED:            23031  23029     -      -      -      -      -      -      -      1      1
                               92.8%  93.5%                                                  4.5%   0.6%
    NEW NUMBERS                20955  20954     -      -      -      -      -      -      -      -      1
                               84.4%  85.0%                                                         0.6%
    DAY PART 1 ONLY              113    112     -      -      -      -      -      -      -      1      -
                                0.5%   0.5%                                                  4.5%
    DAY PART 2 ONLY               67     67     -      -      -      -      -      -      -      -      -
                                0.3%   0.3%
    DAY PART 3 ONLY               61     61     -      -      -      -      -      -      -      -      -
                                0.2%   0.2%
    DAY PART 1 OR 2              592    592     -      -      -      -      -      -      -      -      -
                                2.4%   2.4%
    DAY PART 1 OR 3              643    643     -      -      -      -      -      -      -      -      -
                                2.6%   2.6%
    DAY PART 2 OR 3              552    552     -      -      -      -      -      -      -      -      -
                                2.2%   2.2%
    DAY PART 1, 2, OR 3            3      3     -      -      -      -      -      -      -      -      -
                                   *      *
    DAY PART TARGETS MET          45     45     -      -      -      -      -      -      -      -      -
                                0.2%   0.2%

  TIMED CALLBACKS TODAY         1616   1613     -      -      -      -      -      -      -      2      1
                                6.5%   6.5%                                                  9.1%   0.6%

  ***TIMED MORNING (6-<12)       262    262     -      -      -      -      -      -      -      -      -
                                1.1%   1.1%

  ***TIMED AFTERNOON (12-<6)     686    686     -      -      -      -      -      -      -      -      -
                                2.8%   2.8%

  ***TIMED EVENING (6-<12)       141    141     -      -      -      -      -      -      -      -      -
                                0.6%   0.6%

  TIMED LATER                    527    524     -      -      -      -      -      -      -      2      1
                                2.1%   2.1%                                                  9.1%   0.6%

  SPECIAL INTERVIEWER STACKS     176      -     -      -      -      -      -      -      -     19    157
                                0.7%                                                        86.4%  98.7%

**EASTERN TZ AVAILABLE:        13566  13464     -      -      -      -      -      -      -     13     89
                               54.7%  54.6%                                                 59.1%  56.0%

  SYSTEM SCHEDULED:            12702  12701     -      -      -      -      -      -      -      1      -
                               51.2%  51.5%                                                  4.5%
    NEW NUMBERS                11599  11599     -      -      -      -      -      -      -      -      -
                               46.7%  47.1%
```

```
(continued)
                          REPORT 8:  SPECIAL INTERVIEW TYPE BY ACTIVE NUMBER AVAILABILITY (STACK) WITHIN TIME ZONE

                                           NOT   <================ SPECIAL INTERVIEWER TYPES =================>
                                   TOTAL SPECIAL   TYPE   TYPE   TYPE   TYPE   TYPE   TYPE   TYPE   TYPE   TYPE
TIME ZONE/AVAILABILITY            ACTIVE  TYPE      1      2      3      4      5      6      7      8      9
----------------------            ------ ------ ---- ---- ---- ---- ---- ----   ----   ----   ----   ----

    DAY PART 1 ONLY                 35     34      -      -      -      -      -      -      -      1      -
                                   0.1%   0.1%                                                     4.5%
    DAY PART 2 ONLY                 13     13      -      -      -      -      -      -      -      -      -
                                   0.1%   0.1%
    DAY PART 3 ONLY                 42     42      -      -      -      -      -      -      -      -      -
                                   0.2%   0.2%
    DAY PART 1 OR 2                232    232      -      -      -      -      -      -      -      -      -
                                   0.9%   0.9%
    DAY PART 1 OR 3                410    410      -      -      -      -      -      -      -      -      -
                                   1.7%   1.7%
    DAY PART 2 OR 3                362    362      -      -      -      -      -      -      -      -      -
                                   1.5%   1.5%
    DAY PART 1, 2 OR 3               1      1      -      -      -      -      -      -      -      -      -
                                     *      *
    DAY PART TARGETS MET             8      8      -      -      -      -      -      -      -      -      -
                                     *      *

  TIMED CALLBACKS TODAY            765    763      -      -      -      -      -      -      -      2      -
                                   3.1%   3.1%                                                     9.1%

  ***TIMED MORNING (6-<12)         177    177      -      -      -      -      -      -      -      -      -
                                   0.7%   0.7%

  ***TIMED AFTERNOON (12-<6)       267    267      -      -      -      -      -      -      -      -      -
                                   1.1%   1.1%

  ***TIMED EVENING (6-<12)           1      1      -      -      -      -      -      -      -      -      -
                                     *      *

  TIMED LATER                      320    318      -      -      -      -      -      -      -      2      -
                                   1.3%   1.3%                                                     9.1%

  SPECIAL INTERVIEWER ONLY          99      -      -      -      -      -      -      -      -     10     89
                                   0.4%                                                           45.5%  56.0%

**CENTRAL TZ AVAILABLE:           8688   8638      -      -      -      -      -      -      -      5     45
                                  35.0%  35.1%                                                    22.7%  28.3%

  SYSTEM SCHEDULED:               8089   8088      -      -      -      -      -      -      -      -      1
                                  32.6%  32.8%                                                           0.6%
    NEW NUMBERS                   7324   7323      -      -      -      -      -      -      -      -      1
                                  29.5%  29.7%                                                           0.6%
    DAY PART 1 ONLY                 48     48      -      -      -      -      -      -      -      -      -
                                   0.2%   0.2%
    DAY PART 2 ONLY                 41     41      -      -      -      -      -      -      -      -      -
                                   0.2%   0.2%
    DAY PART 3 ONLY                 19     19      -      -      -      -      -      -      -      -      -
                                   0.1%   0.1%
```

```
(continued)
                        REPORT 8:  SPECIAL INTERVIEW TYPE BY ACTIVE NUMBER AVAILABILITY (STACK) WITHIN TIME ZONE

                                         NOT   <================ SPECIAL INTERVIEWER TYPES =================>
                                TOTAL SPECIAL   TYPE   TYPE   TYPE   TYPE   TYPE   TYPE   TYPE   TYPE   TYPE
TIME ZONE/AVAILABILITY        ACTIVE  TYPE       1      2      3      4      5      6      7      8      9
---------------------          ------ ------ ---- ---- ---- ---- ---- ---- ----   ----   ----   ----   ----

    DAY PART 1 OR 2              225    225      -      -       -      -      -      -      -       -      -
                                 0.9%   0.9%
    DAY PART 1 OR 3              232    232      -      -       -      -      -      -      -       -      -
                                 0.9%   0.9%
    DAY PART 2 OR 3              190    190      -      -       -      -      -      -      -       -      -
                                 0.8%   0.8%
    DAY PART 1, 2, OR 3            2      2      -      -       -      -      -      -      -       -      -
                                   *      *
    DAY PART TARGETS MET           8      8      -      -       -      -      -      -      -       -      -
                                   *      *

  TIMED CALLBACKS TODAY          550    550      -      -       -      -      -      -      -       -      -
                                 2.2%   2.2%

  ***TIMED MORNING (6-<12)        70     70      -      -       -      -      -      -      -       -      -
                                 0.3%   0.3%

  ***TIMED AFTERNOON (12-<6)     298    298      -      -       -      -      -      -      -       -      -
                                 1.2%   1.2%

  ***TIMED EVENING (6-<12)        14     14      -      -       -      -      -      -      -       -      -
                                 0.1%   0.1%

  TIMED LATER                    168    168      -      -       -      -      -      -      -       -      -
                                 0.7%   0.7%

  SPECIAL INTERVIEWER ONLY        49      -      -      -       -      -      -      -      -       5     44
                                 0.2%                                                          22.7%  27.7%

**MOUNTAIN TZ AVAILABLE:         530    523      -      -       -      -      -      -      -       1      6
                                 2.1%   2.1%                                                    4.5%   3.8%

  SYSTEM SCHEDULED:              461    461      -      -       -      -      -      -      -       -      -
                                 1.9%   1.9%
    NEW NUMBERS                  418    418      -      -       -      -      -      -      -       -      -
                                 1.7%   1.7%
    DAY PART 2 ONLY                7      7      -      -       -      -      -      -      -       -      -
                                   *      *
    DAY PART 1 OR 2               28     28      -      -       -      -      -      -      -       -      -
                                 0.1%   0.1%
    DAY PART TARGETS MET           8      8      -      -       -      -      -      -      -       -      -
                                   *      *

  TIMED CALLBACKS TODAY           62     62      -      -       -      -      -      -      -       -      -
                                 0.2%   0.3%

  ***TIMED MORNING (6-<12)        12     12      -      -       -      -      -      -      -       -      -
                                   *      *
```

```
(continued)
                        REPORT 8:  SPECIAL INTERVIEW TYPE BY ACTIVE NUMBER AVAILABILITY (STACK) WITHIN TIME ZONE

                                          NOT   <================ SPECIAL INTERVIEWER TYPES =================>
                                   TOTAL SPECIAL   TYPE    TYPE    TYPE    TYPE    TYPE    TYPE    TYPE    TYPE    TYPE
TIME ZONE/AVAILABILITY            ACTIVE  TYPE      1       2       3       4       5       6       7       8       9
----------------------             ------ ------ ---- ---- ---- ---- ---- ---- ----    ----    ----    ----    ----

  ***TIMED AFTERNOON (12-<6)        22     22      -       -       -       -       -       -       -       -       -
                                   0.1%   0.1%

  ***TIMED EVENING (6-<12)          15     15      -       -       -       -       -       -       -       -       -
                                   0.1%   0.1%

  TIMED LATER                       13     13      -       -       -       -       -       -       -       -       -
                                   0.1%   0.1%

  SPECIAL INTERVIEWER ONLY           7      -      -       -       -       -       -       -       -       1       6
                                     *                                                               4.5%    3.8%

**PACIFIC TZ AVAILABLE:           2039   2017      -       -       -       -       -       -       -       3      19
                                   8.2%   8.2%                                                      13.6%   11.9%

  SYSTEM SCHEDULED:               1779   1779      -       -       -       -       -       -       -       -       -
                                   7.2%   7.2%
    NEW NUMBERS                   1614   1614      -       -       -       -       -       -       -       -       -
                                   6.5%   6.5%
    DAY PART 1 ONLY                 30     30      -       -       -       -       -       -       -       -       -
                                   0.1%   0.1%
    DAY PART 2 ONLY                  6      6      -       -       -       -       -       -       -       -       -
                                     *      *
    DAY PART 1 OR 2                107    107      -       -       -       -       -       -       -       -       -
                                   0.4%   0.4%
    DAY PART 1 OR 3                  1      1      -       -       -       -       -       -       -       -       -
                                     *      *
    DAY PART TARGETS MET            21     21      -       -       -       -       -       -       -       -       -
                                   0.1%   0.1%

  TIMED CALLBACKS TODAY            239    238      -       -       -       -       -       -       -       -       1
                                   1.0%   1.0%                                                               0.6%

  ***TIMED MORNING (6-<12)           3      3      -       -       -       -       -       -       -       -       -
                                     *      *

  ***TIMED AFTERNOON (12-<6)        99     99      -       -       -       -       -       -       -       -       -
                                   0.4%   0.4%

  ***TIMED EVENING (6-<12)         111    111      -       -       -       -       -       -       -       -       -
                                   0.4%   0.5%

  TIMED LATER                       26     25      -       -       -       -       -       -       -       -       1
                                   0.1%   0.1%                                                               0.6%

  SPECIAL INTERVIEWER ONLY          21      -      -       -       -       -       -       -       -       3      18
                                   0.1%                                                             13.6%   11.3%

Note: Percentage less than 0.05 printed as *.
```

```
                         Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 28

                                     REPORT 9:  NUMBER OF CALLS MADE BY AVAILABILITY WITHIN MARKET

                                         NOT <---------------------  NUMBER OF CALLS MADE  ------------------------>           AVERAGE
                                TOTAL    YET                                                                 10 OR     TOTAL # CALLS
MARKET AVAILABILITY           NUMBERS CALLED    ONE    TWO  THREE   FOUR   FIVE    SIX  SEVEN  EIGHT   NINE   MORE     CALLS    MADE
-------------------           ------- ------  -----  -----  -----  ------ ------  -----  -----  ------ ------ ------   -------  ------

TOTAL SAMPLE:                  28130  20958   3653   2100    938    323    109     35     10      3      -      1      12818    0.46
-------------                  100.0% 100.0% 100.0% 100.0% 100.0% 100.0% 100.0% 100.0% 100.0% 100.0%        100.0%    100.0%

  RESOLVED NUMBERS:             3307      -   2160    743    241     96     40     19      6      2      -      -       5125    1.55
                                11.8%          59.1%  35.4%  25.7%  29.7%  36.7%  54.3%  60.0%  66.7%                    40.0%
    COMPLETES                     86      -      1     14     39     21      9      2      -      -      -      -        287    3.34
                                 0.3%              *   0.7%   4.2%   6.5%   8.3%   5.7%                                   2.2%
    OTHER RESOLVED              3221      -   2159    729    202     75     31     17      6      2      -      -       4838    1.50
                                11.5%          59.1%  34.7%  21.5%  23.2%  28.4%  48.6%  60.0%  66.7%                    37.7%

  AVAILABLE NUMBERS:           24823  20958   1493   1357    697    227     69     16      4      1      -      1       7693    0.31
                                88.2% 100.0%  40.9%  64.6%  74.3%  70.3%  63.3%  45.7%  40.0%  33.3%        100.0%    60.0%
    SYSTEM SCHEDULED:          23031  20955   1060    672    254     73     15      2      -      -      -      -       3545    0.15
                                81.9% 100.0%  29.0%  32.0%  27.1%  22.6%  13.8%   5.7%                                  27.7%
      TODAY (BUCKET 0-7)       22986  20955   1060    672    235     55      8      1      -      -      -      -       3375    0.15
                                81.7% 100.0%  29.0%  32.0%  25.1%  17.0%   7.3%   2.9%                                  26.3%
    TIMED TODAY                 1089      -    242    464    263     84     28      7      1      -      -      -       2484    2.28
                                 3.9%           6.6%  22.1%  28.0%  26.0%  25.7%  20.0%  10.0%                           19.4%
    TIMED LATER                  527      2    167    164    117     50     18      5      2      1      -      1       1198    2.27
                                 1.9%      *   4.6%   7.8%  12.5%  15.5%  16.5%  14.3%  20.0%  33.3%        100.0%     9.3%
    SPCL INTV/HIDDEN             176      1     24     57     63     20      8      2      1      -      -      -        466    2.65
                                 0.6%      *   0.7%   2.7%   6.7%   6.2%   7.3%   5.7%  10.0%                            3.6%

TOTAL MARKET 01:                2120   1778    151    116     51     21      2      1      -      -      -      -        636    0.30
----------------                 7.5%   8.5%   4.1%   5.5%   5.4%   6.5%   1.8%   2.9%                                   5.0%

  RESOLVED NUMBERS:              132      -     72     42     13      5      -      -      -      -      -      -        215    1.63
                                 0.5%           2.0%   2.0%   1.4%   1.5%                                                 1.7%
    COMPLETES                      8      -      -      2      4      2      -      -      -      -      -      -         24    3.00
                                    *                  0.1%   0.4%   0.6%                                                 0.2%
    OTHER RESOLVED               124      -     72     40      9      3      -      -      -      -      -      -        191    1.54
                                 0.4%           2.0%   1.9%   1.0%   0.9%                                                 1.5%

  AVAILABLE NUMBERS:            1988   1778     79     74     38     16      2      1      -      -      -      -        421    0.21
                                 7.1%   8.5%   2.2%   3.5%   4.1%   5.0%   1.8%   2.9%                                   3.3%
    SYSTEM SCHEDULED:           1910   1778     65     45     15      6      1      -      -      -      -      -        229    0.12
                                 6.8%   8.5%   1.8%   2.1%   1.6%   1.9%   0.9%                                          1.8%
      TODAY (BUCKET 0-7)        1910   1778     65     45     15      6      1      -      -      -      -      -        229    0.12
                                 6.8%   8.5%   1.8%   2.1%   1.6%   1.9%   0.9%                                          1.8%
    TIMED TODAY                   46      -      9     19     11      7      -      -      -      -      -      -        108    2.35
                                 0.2%           0.2%   0.9%   1.2%   2.2%                                                 0.8%
    TIMED LATER                   17      -      4      5      5      3      -      -      -      -      -      -         41    2.41
                                 0.1%           0.1%   0.2%   0.5%   0.9%                                                 0.3%
    SPCL INTV/HIDDEN              15      -      1      5      7      -      1      1      -      -      -      -         43    2.87
                                 0.1%              *   0.2%   0.7%          0.9%   2.9%                                   0.3%

TOTAL MARKET 02:                4195   3104    597    326    121     36      8      3      -      -      -      -       1814    0.43
----------------                14.9%  14.8%  16.3%  15.5%  12.9%  11.1%   7.3%   8.6%                                  14.2%
```

```
                                  Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 29

(continued)
                                                    REPORT 9:  NUMBER OF CALLS MADE BY AVAILABILITY WITHIN MARKET

                                            NOT   <---------------------  NUMBER OF CALLS MADE  ---------------------->          AVERAGE
                                    TOTAL   YET                                                                    10 OR     TOTAL # CALLS
MARKET AVAILABILITY               NUMBERS CALLED   ONE    TWO   THREE  FOUR   FIVE    SIX   SEVEN  EIGHT  NINE   MORE      CALLS   MADE
-------------------               ------- ------  -----  -----  ----- ------ ------  -----  -----  ------ ------ ------    ------- ------

  RESOLVED NUMBERS:                 484     -     333    106     32     10      2      1      -      -      -      -         697    1.44
                                    1.7%          9.1%   5.0%   3.4%   3.1%   1.8%   2.9%                                    5.4%
    COMPLETES                         9     -       -      1      4      2      2      -      -      -      -      -          32    3.56
                                      *                    *    0.4%   0.6%   1.8%                                           0.2%
    OTHER RESOLVED                  475     -     333    105     28      8      -      1      -      -      -      -         665    1.40
                                    1.7%          9.1%   5.0%   3.0%   2.5%          2.9%                                    5.2%

  AVAILABLE NUMBERS:               3711   3104    264    220     89     26      6      2      -      -      -      -        1117    0.30
                                   13.2%  14.8%   7.2%  10.5%   9.5%   8.0%   5.5%   5.7%                                    8.7%
    SYSTEM SCHEDULED:              3445   3103    181    116     36      7      2      -      -      -      -      -         559    0.16
                                   12.2%  14.8%   5.0%   5.5%   3.8%   2.2%   1.8%                                           4.4%
      TODAY (BUCKET 0-7)           3443   3103    181    116     36      6      1      -      -      -      -      -         550    0.16
                                   12.2%  14.8%   5.0%   5.5%   3.8%   1.9%   0.9%                                           4.3%
    TIMED TODAY                     160     -      44     65     35     13      2      1      -      -      -      -         347    2.17
                                    0.6%          1.2%   3.1%   3.7%   4.0%   1.8%   2.9%                                    2.7%
    TIMED LATER                      80     -      37     27     11      3      1      1      -      -      -      -         147    1.84
                                    0.3%          1.0%   1.3%   1.2%   0.9%   0.9%   2.9%                                    1.1%
    SPCL INTV/HIDDEN                 26     1       2     12      7      3      1      -      -      -      -      -          64    2.46
                                    0.1%     *    0.1%   0.6%   0.7%   0.9%   0.9%                                           0.5%

TOTAL MARKET 03:                   6307   4782    831    435    191     48     14      3      2      -      -      1        2578    0.41
----------------                   22.4%  22.8%  22.7%  20.7%  20.4%  14.9%  12.8%   8.6%  20.0%                100.0%      20.1%

  RESOLVED NUMBERS:                 712     -     484    158     44     13      8      3      2      -      -      -        1056    1.48
                                    2.5%         13.2%   7.5%   4.7%   4.0%   7.3%   8.6%  20.0%                             8.2%
    COMPLETES                        16     -       -      2      8      2      3      1      -      -      -      -          57    3.56
                                    0.1%                 0.1%   0.9%   0.6%   2.8%   2.9%                                    0.4%
    OTHER RESOLVED                  696     -     484    156     36     11      5      2      2      -      -      -         999    1.44
                                    2.5%         13.2%   7.4%   3.8%   3.4%   4.6%   5.7%  20.0%                             7.8%

  AVAILABLE NUMBERS:               5595   4782    347    277    147     35      6      -      -      -      -      1        1522    0.27
                                   19.9%  22.8%   9.5%  13.2%  15.7%  10.8%   5.5%                              100.0%      11.9%
    SYSTEM SCHEDULED:              5241   4782    241    150     50     17      1      -      -      -      -      -         764    0.15
                                   18.6%  22.8%   6.6%   7.1%   5.3%   5.3%   0.9%                                           6.0%
      TODAY (BUCKET 0-7)           5239   4782    241    150     49     17      -      -      -      -      -      -         756    0.14
                                   18.6%  22.8%   6.6%   7.1%   5.2%   5.3%                                                  5.9%
    TIMED TODAY                     228     -      62     96     58      9      3      -      -      -      -      -         479    2.10
                                    0.8%          1.7%   4.6%   6.2%   2.8%   2.8%                                           3.7%
    TIMED LATER                     104     -      38     26     31      7      1      -      -      -      -      1         226    2.17
                                    0.4%          1.0%   1.2%   3.3%   2.2%   0.9%                              100.0%       1.8%
    SPCL INTV/HIDDEN                 22     -       6      5      8      2      1      -      -      -      -      -          53    2.41
                                    0.1%          0.2%   0.2%   0.9%   0.6%   0.9%                                           0.4%

TOTAL MARKET 04:                   1374    914    213    109     87     33     11      7      -      -      -      -         921    0.67
----------------                    4.9%   4.4%   5.8%   5.2%   9.3%  10.2%  10.1%  20.0%                                    7.2%

  RESOLVED NUMBERS:                 235     -     155     44     20      9      4      3      -      -      -      -         377    1.60
                                    0.8%          4.2%   2.1%   2.1%   2.8%   3.7%   8.6%                                    2.9%
```

```
                         Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 30

(continued)
                                     REPORT 9:  NUMBER OF CALLS MADE BY AVAILABILITY WITHIN MARKET

                                        NOT    <---------------------  NUMBER OF CALLS MADE  ----------------------->           AVERAGE
                               TOTAL   YET                                                                     10 OR     TOTAL # CALLS
MARKET AVAILABILITY           NUMBERS CALLED   ONE    TWO   THREE  FOUR   FIVE    SIX    SEVEN  EIGHT  NINE   MORE     CALLS   MADE
-------------------           ------- ------   -----  -----  ----- ------ ------   -----  -----  ------ ------ ------    ------- ------

    COMPLETES                      6      -      1       2      1      1      1      -       -     -      -       -        17     2.83
                                   *             *    0.1%   0.1%   0.3%   0.9%                                             0.1%
    OTHER RESOLVED               229      -    154      42     19      8      3      3       -     -      -       -       360     1.57
                                0.8%          4.2%    2.0%   2.0%   2.5%   2.8%   8.6%                                      2.8%

  AVAILABLE NUMBERS:            1139    914     58      65     67     24      7      4       -     -      -       -       544     0.48
                                4.0%   4.4%   1.6%    3.1%   7.1%   7.4%   6.4%  11.4%                                      4.2%
    SYSTEM SCHEDULED:            999    914     30      23     24      6      1      1       -     -      -       -       183     0.18
                                3.6%   4.4%   0.8%    1.1%   2.6%   1.9%   0.9%   2.9%                                      1.4%
      TODAY (BUCKET 0-7)         983    914     30      23     13      2      -      1       -     -      -       -       129     0.13
                                3.5%   4.4%   0.8%    1.1%   1.4%   0.6%          2.9%                                      1.0%
    TIMED TODAY                  112      -     27      33     31     13      5      3       -     -      -       -       281     2.51
                                0.4%          0.7%    1.6%   3.3%   4.0%   4.6%   8.6%                                      2.2%
    TIMED LATER                   17      -      -       6      8      3      -      -       -     -      -       -        48     2.82
                                0.1%                  0.3%   0.9%   0.9%                                                    0.4%
    SPCL INTV/HIDDEN              11      -      1       3      4      2      1      -       -     -      -       -        32     2.91
                                   *             *    0.1%   0.4%   0.6%   0.9%                                             0.2%

TOTAL MARKET 05:                3327   2554    402     262     85     23      1      -       -     -      -       -      1278     0.38
----------------               11.8%  12.2%  11.0%   12.5%   9.1%   7.1%   0.9%                                            10.0%

  RESOLVED NUMBERS:              323      -    210      88     17      8      -      -       -     -      -       -       469     1.45
                                1.1%          5.7%    4.2%   1.8%   2.5%                                                    3.7%
    COMPLETES                      4      -      -       -      2      2      -      -       -     -      -       -        14     3.50
                                   *                           0.2%   0.6%                                                    0.1%
    OTHER RESOLVED               319      -    210      88     15      6      -      -       -     -      -       -       455     1.43
                                1.1%          5.7%    4.2%   1.6%   1.9%                                                    3.5%

  AVAILABLE NUMBERS:            3004   2554    192     174     68     15      1      -       -     -      -       -       809     0.27
                               10.7%  12.2%   5.3%    8.3%   7.2%   4.6%   0.9%                                             6.3%
    SYSTEM SCHEDULED:           2829   2552    150      93     27      6      1      -       -     -      -       -       446     0.16
                               10.1%  12.2%   4.1%    4.4%   2.9%   1.9%   0.9%                                             3.5%
      TODAY (BUCKET 0-7)        2828   2552    150      93     27      5      1      -       -     -      -       -       442     0.16
                               10.1%  12.2%   4.1%    4.4%   2.9%   1.5%   0.9%                                             3.4%
    TIMED TODAY                   99      -     24      47     24      4      -      -       -     -      -       -       206     2.08
                                0.4%          0.7%    2.2%   2.6%   1.2%                                                    1.6%
    TIMED LATER                   56      2     15      27      8      4      -      -       -     -      -       -       109     1.95
                                0.2%      *   0.4%    1.3%   0.9%   1.2%                                                    0.9%
    SPCL INTV/HIDDEN              20      -      3       7      9      1      -      -       -     -      -       -        48     2.40
                                0.1%          0.1%    0.3%   1.0%   0.3%                                                    0.4%

TOTAL MARKET 06:                3684   2745    538     259    109     22     10      1       -     -      -       -      1527     0.41
----------------               13.1%  13.1%  14.7%   12.3%  11.6%   6.8%   9.2%   2.9%                                     11.9%

  RESOLVED NUMBERS:              456      -    312      95     34     10      5      -       -     -      -       -       669     1.47
                                1.6%          8.5%    4.5%   3.6%   3.1%   4.6%                                             5.2%
    COMPLETES                     14      -      -       2      7      4      1      -       -     -      -       -        46     3.29
                                   *                  0.1%   0.7%   1.2%   0.9%                                             0.4%
                         Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 31

(continued)
                                     REPORT 9:  NUMBER OF CALLS MADE BY AVAILABILITY WITHIN MARKET

                                         NOT   <---------------------  NUMBER OF CALLS MADE  ----------------------->          AVERAGE
                               TOTAL   YET                                                                     10 OR     TOTAL # CALLS
MARKET AVAILABILITY           NUMBERS CALLED   ONE     TWO   THREE  FOUR   FIVE     SIX   SEVEN  EIGHT  NINE   MORE      CALLS   MADE
-------------------           ------- ------   -----  -----  -----  ------  ------  -----  -----  ------ ------ ------   ------- ------

    OTHER RESOLVED               442      -    312     93     27       6      4      -       -      -      -       -       623    1.41
                                 1.6%           8.5%   4.4%   2.9%   1.9%   3.7%                                            4.9%

  AVAILABLE NUMBERS:            3228   2745    226    164     75      12      5      1       -      -      -       -       858    0.27
                                11.5%  13.1%    6.2%   7.8%   8.0%   3.7%   4.6%   2.9%                                     6.7%
    SYSTEM SCHEDULED:           3016   2745    160     76     29       5      1      -       -      -      -       -       424    0.14
                                10.7%  13.1%    4.4%   3.6%   3.1%   1.5%   0.9%                                            3.3%
      TODAY (BUCKET 0-7)        3016   2745    160     76     29       5      1      -       -      -      -       -       424    0.14
                                10.7%  13.1%    4.4%   3.6%   3.1%   1.5%   0.9%                                            3.3%
    TIMED TODAY                  126      -     32     60     26       5      3      -       -      -      -       -       265    2.10
                                 0.4%           0.9%   2.9%   2.8%   1.5%   2.8%                                            2.1%
    TIMED LATER                   63      -     29     22     10       1      -      1       -      -      -       -       113    1.79
                                 0.2%           0.8%   1.0%   1.1%   0.3%          2.9%                                     0.9%
    SPCL INTV/HIDDEN              23      -      5      6     10       1      1      -       -      -      -       -        56    2.43
                                 0.1%           0.1%   0.3%   1.1%   0.3%   0.9%                                            0.4%

TOTAL MARKET 07:                5002   3972    559    315    102      34     15      4       -      1      -       -      1738    0.35
----------------                17.8%  19.0%   15.3%  15.0%  10.9%  10.5%  13.8%  11.4%          33.3%                    13.6%

  RESOLVED NUMBERS:              494      -    347    102     23      10      8      3       -      1      -       -       726    1.47
                                 1.8%           9.5%   4.9%   2.5%   3.1%   7.3%   8.6%           33.3%                     5.7%
    COMPLETES                     10      -      -      2      5       2      1      -       -      -      -       -        32    3.20
                                   *                   0.1%   0.5%   0.6%   0.9%                                            0.2%
    OTHER RESOLVED               484      -    347    100     18       8      7      3       -      1      -       -       694    1.43
                                 1.7%           9.5%   4.8%   1.9%   2.5%   6.4%   8.6%           33.3%                     5.4%

  AVAILABLE NUMBERS:            4508   3972    212    213     79      24      7      1       -      -      -       -      1012    0.22
                                16.0%  19.0%    5.8%  10.1%   8.4%   7.4%   6.4%   2.9%                                     7.9%
    SYSTEM SCHEDULED:           4271   3972    151     99     36      11      2      -       -      -      -       -       511    0.12
                                15.2%  19.0%    4.1%   4.7%   3.8%   3.4%   1.8%                                            4.0%
      TODAY (BUCKET 0-7)        4267   3972    151     99     36       9      -      -       -      -      -       -       493    0.12
                                15.2%  19.0%    4.1%   4.7%   3.8%   2.8%                                                   3.8%
    TIMED TODAY                  143      -     24     76     30       9      3      1       -      -      -       -       323    2.26
                                 0.5%           0.7%   3.6%   3.2%   2.8%   2.8%   2.9%                                     2.5%
    TIMED LATER                   71      -     34     28      6       1      2      -       -      -      -       -       122    1.72
                                 0.3%           0.9%   1.3%   0.6%   0.3%   1.8%                                            1.0%
    SPCL INTV/HIDDEN              23      -      3     10      7       3      -      -       -      -      -       -        56    2.43
                                 0.1%           0.1%   0.5%   0.7%   0.9%                                                   0.4%

TOTAL MARKET 08:                1599   1074    202    169     87      42     20      4       -      1      -       -      1101    0.69
----------------                 5.7%   5.1%    5.5%   8.0%   9.3%  13.0%  18.3%  11.4%          33.3%                     8.6%

  RESOLVED NUMBERS:              267      -    144     68     28      18      5      3       -      1      -       -       487    1.82
                                 0.9%           3.9%   3.2%   3.0%   5.6%   4.6%   8.6%           33.3%                     3.8%
    COMPLETES                     12      -      -      1      5       5      -      1       -      -      -       -        43    3.58
                                   *                     *    0.5%   1.5%          2.9%                                     0.3%
    OTHER RESOLVED               255      -    144     67     23      13      5      2       -      1      -       -       444    1.74
                                 0.9%           3.9%   3.2%   2.5%   4.0%   4.6%   5.7%           33.3%                     3.5%
```

```
(continued)
                                     REPORT 9:  NUMBER OF CALLS MADE BY AVAILABILITY WITHIN MARKET

                                          NOT  <---------------------  NUMBER OF CALLS MADE  --------------------->           AVERAGE
                              TOTAL    YET                                                                  10 OR      TOTAL # CALLS
MARKET AVAILABILITY         NUMBERS CALLED    ONE    TWO  THREE   FOUR   FIVE    SIX  SEVEN  EIGHT   NINE   MORE      CALLS   MADE
-------------------         ------- ------  -----  -----  ----- ------ ------  -----  ----- ------ ------ ------    ------- ------

  AVAILABLE NUMBERS:           1332   1074     58    101     59     24     15      1      -      -      -      -        614   0.46
                               4.7%   5.1%   1.6%   4.8%   6.3%   7.4%  13.8%   2.9%                                    4.8%
    SYSTEM SCHEDULED:          1175   1074     44     35     15      3      4      -      -      -      -      -        191   0.16
                               4.2%   5.1%   1.2%   1.7%   1.6%   0.9%   3.7%                                           1.5%
      TODAY (BUCKET 0-7)       1166   1074     44     35     11      -      2      -      -      -      -      -        157   0.13
                               4.1%   5.1%   1.2%   1.7%   1.2%          1.8%                                           1.2%
    TIMED TODAY                 125      -     13     58     29     16      9      -      -      -      -      -        325   2.60
                               0.4%          0.4%   2.8%   3.1%   5.0%   8.3%                                           2.5%
    TIMED LATER                  20      -      -      4     11      3      1      1      -      -      -      -         64   3.20
                               0.1%                 0.2%   1.2%   0.9%   0.9%   2.9%                                    0.5%
    SPCL INTV/HIDDEN             12      -      1      4      4      2      1      -      -      -      -      -         34   2.83
                                  *             *   0.2%   0.4%   0.6%   0.9%                                           0.3%

TOTAL MARKET 09:                122      -     23     15     34     26     14      3      6      1      -      -        397   3.25
----------------               0.4%          0.6%   0.7%   3.6%   8.0%  12.8%   8.6%  60.0%  33.3%                      3.1%

  RESOLVED NUMBERS:              52      -     23     11      8      3      4      1      2      -      -      -        121   2.33
                               0.2%          0.6%   0.5%   0.9%   0.9%   3.7%   2.9%  20.0%                             0.9%
    COMPLETES                     1      -      -      -      1      -      -      -      -      -      -      -          3   3.00
                                  *                         0.1%                                                           *
    OTHER RESOLVED               51      -     23     11      7      3      4      1      2      -      -      -        118   2.31
                               0.2%          0.6%   0.5%   0.7%   0.9%   3.7%   2.9%  20.0%                             0.9%

  AVAILABLE NUMBERS:             70      -      -      4     26     23     10      2      4      1      -      -        276   3.94
                               0.2%                 0.2%   2.8%   7.1%   9.2%   5.7%  40.0%  33.3%                      2.2%
    SYSTEM SCHEDULED:            14      -      -      2      6      5      -      1      -      -      -      -         48   3.43
                                  *                 0.1%   0.6%   1.5%          2.9%                                    0.4%
      TODAY (BUCKET 0-7)          9      -      -      2      5      2      -      -      -      -      -      -         27   3.00
                                  *                 0.1%   0.5%   0.6%                                                  0.2%
    TIMED TODAY                   9      -      -      -      4      1      3      -      1      -      -      -         38   4.22
                                  *                        0.4%   0.3%   2.8%         10.0%                             0.3%
    TIMED LATER                  39      -      -      2     13     14      6      1      2      1      -      -        157   4.03
                               0.1%                 0.1%   1.4%   4.3%   5.5%   2.9%  20.0%  33.3%                      1.2%
    SPCL INTV/HIDDEN              8      -      -      -      3      3      1      -      1      -      -      -         33   4.13
                                  *                        0.3%   0.9%   0.9%         10.0%                             0.3%

TOTAL MARKET 10:                120      -     46     28     25     14      4      3      -      -      -      -        271   2.26
----------------               0.4%          1.3%   1.3%   2.7%   4.3%   3.7%   8.6%                                    2.1%

  RESOLVED NUMBERS:              54      -     28      7     10      6      -      3      -      -      -      -        114   2.11
                               0.2%          0.8%   0.3%   1.1%   1.9%          8.6%                                    0.9%
    COMPLETES                     1      -      -      -      1      -      -      -      -      -      -      -          3   3.00
                                  *                         0.1%                                                           *
    OTHER RESOLVED               53      -     28      7      9      6      -      3      -      -      -      -        111   2.09
                               0.2%          0.8%   0.3%   1.0%   1.9%          8.6%                                    0.9%

  AVAILABLE NUMBERS:             66      -     18     21     15      8      4      -      -      -      -      -        157   2.38
                               0.2%          0.5%   1.0%   1.6%   2.5%   3.7%                                           1.2%
```

```
                        Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 33

(continued)
                                    REPORT 9:  NUMBER OF CALLS MADE BY AVAILABILITY WITHIN MARKET

                                        NOT   <--------------------  NUMBER OF CALLS MADE  --------------------->          AVERAGE
                              TOTAL   YET                                                                     10 OR    TOTAL # CALLS
MARKET AVAILABILITY          NUMBERS CALLED   ONE    TWO   THREE  FOUR   FIVE    SIX   SEVEN  EIGHT  NINE   MORE     CALLS   MADE
-------------------          ------- ------  -----  -----  ----- ------ ------  -----  ----- ------ ------ ------   ------- ------

    SYSTEM SCHEDULED:            33      -     18     11      2      2      -      -      -      -      -      -        54    1.64
                                0.1%          0.5%   0.5%   0.2%   0.6%                                                0.4%
      TODAY (BUCKET 0-7)         32      -     18     11      2      1      -      -      -      -      -      -        50    1.56
                                0.1%          0.5%   0.5%   0.2%   0.3%                                                0.4%
    TIMED TODAY                   7      -      -      3      4      -      -      -      -      -      -      -        18    2.57
                                   *                 0.1%   0.4%                                                       0.1%
    TIMED LATER                  20      -      -      5      7      4      4      -      -      -      -      -        67    3.35
                                0.1%                 0.2%   0.7%   1.2%   3.7%                                         0.5%
    SPCL INTV/HIDDEN              6      -      -      2      2      2      -      -      -      -      -      -        18    3.00
                                   *                 0.1%   0.2%   0.6%                                                0.1%

TOTAL MARKET 11:                197     35     72     51     22     13      3      1      -      -      -      -       313    1.59
----------------                0.7%   0.2%   2.0%   2.4%   2.3%   4.0%   2.8%   2.9%                                  2.4%

  RESOLVED NUMBERS:              57      -     34     14      6      3      -      -      -      -      -      -        92    1.61
                                0.2%          0.9%   0.7%   0.6%   0.9%                                                0.7%
    COMPLETES                     2      -      -      2      -      -      -      -      -      -      -      -         4    2.00
                                   *                 0.1%                                                                 *
    OTHER RESOLVED               55      -     34     12      6      3      -      -      -      -      -      -        88    1.60
                                0.2%          0.9%   0.6%   0.6%   0.9%                                                0.7%

  AVAILABLE NUMBERS:            140     35     38     37     16     10      3      1      -      -      -      -       221    1.58
                                0.5%   0.2%   1.0%   1.8%   1.7%   3.1%   2.8%   2.9%                                  1.7%
    SYSTEM SCHEDULED:            85     35     20     20      7      3      -      -      -      -      -      -        93    1.09
                                0.3%   0.2%   0.5%   1.0%   0.7%   0.9%                                                0.7%
      TODAY (BUCKET 0-7)         84     35     20     20      7      2      -      -      -      -      -      -        89    1.06
                                0.3%   0.2%   0.5%   1.0%   0.7%   0.6%                                                0.7%
    TIMED TODAY                  12      -      6      2      3      1      -      -      -      -      -      -        23    1.92
                                   *          0.2%   0.1%   0.3%   0.3%                                                0.2%
    TIMED LATER                  37      -     10     12      6      6      2      1      -      -      -      -        92    2.49
                                0.1%          0.3%   0.6%   0.6%   1.9%   1.8%   2.9%                                  0.7%
    SPCL INTV/HIDDEN              6      -      2      3      -      -      1      -      -      -      -      -        13    2.17
                                   *          0.1%   0.1%                 0.9%                                         0.1%

TOTAL MARKET 12:                 83      -     19     15     24     11      7      5      2      -      -      -       244    2.94
----------------                0.3%          0.5%   0.7%   2.6%   3.4%   6.4%  14.3%  20.0%                           1.9%

  RESOLVED NUMBERS:              41      -     18      8      6      1      4      2      2      -      -      -       102    2.49
                                0.1%          0.5%   0.4%   0.6%   0.3%   3.7%   5.7%  20.0%                           0.8%
    COMPLETES                     3      -      -      -      1      1      1      -      -      -      -      -        12    4.00
                                   *                        0.1%   0.3%   0.9%                                         0.1%
    OTHER RESOLVED               38      -     18      8      5      -      3      2      2      -      -      -        90    2.37
                                0.1%          0.5%   0.4%   0.5%          2.8%   5.7%  20.0%                           0.7%

  AVAILABLE NUMBERS:             42      -      1      7     18     10      3      3      -      -      -      -       142    3.38
                                0.1%             *   0.3%   1.9%   3.1%   2.8%   8.6%                                  1.1%
    SYSTEM SCHEDULED:            13      -      -      2      7      2      2      -      -      -      -      -        43    3.31
                                   *                 0.1%   0.7%   0.6%   1.8%                                         0.3%
```

```
                         Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 34

(continued)
                                     REPORT 9:  NUMBER OF CALLS MADE BY AVAILABILITY WITHIN MARKET

                                         NOT   <---------------------- NUMBER OF CALLS MADE  ----------------------->          AVERAGE
                               TOTAL   YET                                                                     10 OR     TOTAL # CALLS
MARKET AVAILABILITY           NUMBERS CALLED   ONE    TWO    THREE  FOUR   FIVE    SIX    SEVEN  EIGHT  NINE   MORE     CALLS    MADE
-------------------           ------- ------   -----  -----  ----- ------ ------   -----  ----- ------ ------ ------    ------- ------

      TODAY (BUCKET 0-7)           9      -      -       2      5      -      2       -      -      -      -       -       29     3.22
                                    *                  0.1%   0.5%          1.8%                                          0.2%
    TIMED TODAY                   22      -       1      5      8      6      -       2      -      -      -       -       71     3.23
                                 0.1%             *    0.2%   0.9%   1.9%          5.7%                                   0.6%
    TIMED LATER                    3      -      -       -      1      1      1       -      -      -      -       -       12     4.00
                                    *                         0.1%   0.3%   0.9%                                          0.1%
    SPCL INTV/HIDDEN               4      -      -       -      2      1      -       1      -      -      -       -       16     4.00
                                    *                         0.2%   0.3%          2.9%                                   0.1%
```

Note: Percentage less than 0.05 printed as *.

```
                         Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 35

                                    REPORT 10:  NUMBER OF CALLS MADE BY AVAILABILITY WITHIN REPLICATE

                                         NOT   <---------------------  NUMBER OF CALLS MADE  ---------------------->          AVERAGE
                               TOTAL   YET                                                                  10 OR     TOTAL # CALLS
REPLICATE AVAILABILITY        NUMBERS CALLED   ONE    TWO    THREE  FOUR   FIVE    SIX   SEVEN  EIGHT  NINE   MORE     CALLS   MADE
----------------------        ------- ------  -----  -----  -----  ------ ------  -----  -----  ------ ------ ------   ------- ------

TOTAL SAMPLE:                  28130  20958   3653   2100    938    323    109     35     10      3      -      1      12818   0.46
-------------                  100.0% 100.0% 100.0% 100.0% 100.0% 100.0% 100.0% 100.0% 100.0% 100.0%         100.0%   100.0%

  RESOLVED NUMBERS:             3307      -   2160    743    241     96     40     19      6      2      -      -       5125   1.55
                               11.8%          59.1%  35.4%  25.7%  29.7%  36.7%  54.3%  60.0%  66.7%                   40.0%
    COMPLETES                     86      -      1     14     39     21      9      2      -      -      -      -        287   3.34
                                0.3%             *    0.7%   4.2%   6.5%   8.3%   5.7%                                  2.2%
    OTHER RESOLVED              3221      -   2159    729    202     75     31     17      6      2      -      -       4838   1.50
                               11.5%          59.1%  34.7%  21.5%  23.2%  28.4%  48.6%  60.0%  66.7%                   37.7%

  AVAILABLE NUMBERS:           24823  20958   1493   1357    697    227     69     16      4      1      -      1       7693   0.31
                               88.2% 100.0%  40.9%  64.6%  74.3%  70.3%  63.3%  45.7%  40.0%  33.3%         100.0%    60.0%
    SYSTEM SCHEDULED:          23031  20955   1060    672    254     73     15      2      -      -      -      -       3545   0.15
                               81.9% 100.0%  29.0%  32.0%  27.1%  22.6%  13.8%   5.7%                                  27.7%
      TODAY (BUCKET 0-7)       22986  20955   1060    672    235     55      8      1      -      -      -      -       3375   0.15
                               81.7% 100.0%  29.0%  32.0%  25.1%  17.0%   7.3%   2.9%                                  26.3%
    TIMED TODAY                 1089      -    242    464    263     84     28      7      1      -      -      -       2484   2.28
                                3.9%           6.6%  22.1%  28.0%  26.0%  25.7%  20.0%  10.0%                          19.4%
    TIMED LATER                  527      2    167    164    117     50     18      5      2      1      -      1       1198   2.27
                                1.9%      *    4.6%   7.8%  12.5%  15.5%  16.5%  14.3%  20.0%  33.3%         100.0%     9.3%
    SPCL INTV/HIDDEN             176      1     24     57     63     20      8      2      1      -      -      -        466   2.65
                                0.6%      *    0.7%   2.7%   6.7%   6.2%   7.3%   5.7%  10.0%                           3.6%

TOTAL REPLICATE 01:             2057     37    785    634    353    154     58     23      9      3      -      1       4253   2.07
-------------------             7.3%   0.2%  21.5%  30.2%  37.6%  47.7%  53.2%  65.7%  90.0% 100.0%         100.0%    33.2%

  RESOLVED NUMBERS:              976      -    560    239     92     43     22     13      5      2      -      -       1725   1.77
                                3.5%          15.3%  11.4%   9.8%  13.3%  20.2%  37.1%  50.0%  66.7%                   13.5%
    COMPLETES                     26      -      -      4     12      7      2      1      -      -      -      -         88   3.38
                                0.1%                  0.2%   1.3%   2.2%   1.8%   2.9%                                  0.7%
    OTHER RESOLVED               950      -    560    235     80     36     20     12      5      2      -      -       1637   1.72
                                3.4%          15.3%  11.2%   8.5%  11.1%  18.3%  34.3%  50.0%  66.7%                   12.8%

  AVAILABLE NUMBERS:            1081     37    225    395    261    111     36     10      4      1      -      1       2528   2.34
                                3.8%   0.2%   6.2%  18.8%  27.8%  34.4%  33.0%  28.6%  40.0%  33.3%         100.0%    19.7%
    SYSTEM SCHEDULED:            578     36    199    211     90     35      6      1      -      -      -      -       1067   1.85
                                2.1%   0.2%   5.4%  10.0%   9.6%  10.8%   5.5%   2.9%                                   8.3%
      TODAY (BUCKET 0-7)         558     36    199    211     83     25      4      -      -      -      -      -        990   1.77
                                2.0%   0.2%   5.4%  10.0%   8.8%   7.7%   3.7%                                          7.7%
    TIMED TODAY                  279      -      9    127     95     31     11      5      1      -      -      -        764   2.74
                                1.0%           0.2%   6.0%  10.1%   9.6%  10.1%  14.3%  10.0%                           6.0%
    TIMED LATER                  166      -     11     43     55     35     15      3      2      1      -      1        527   3.17
                                0.6%           0.3%   2.0%   5.9%  10.8%  13.8%   8.6%  20.0%  33.3%         100.0%     4.1%
    SPCL INTV/HIDDEN              58      1      6     14     21     10      4      1      1      -      -      -        170   2.93
                                0.2%      *    0.2%   0.7%   2.2%   3.1%   3.7%   2.9%  10.0%                           1.3%

TOTAL REPLICATE 02:             1535      -    728    488    224     67     23      5      -      -      -      -       2789   1.82
-------------------             5.5%          19.9%  23.2%  23.9%  20.7%  21.1%  14.3%                                 21.8%
```

```
(continued)
                                 REPORT 10:  NUMBER OF CALLS MADE BY AVAILABILITY WITHIN REPLICATE

                                         NOT   <---------------------  NUMBER OF CALLS MADE  ---------------------->           AVERAGE
                                  TOTAL  YET                                                                 10 OR      TOTAL # CALLS
REPLICATE AVAILABILITY          NUMBERS CALLED   ONE    TWO   THREE  FOUR   FIVE    SIX   SEVEN  EIGHT  NINE   MORE       CALLS   MADE
---------------------           ------- ------  -----  -----  ----- ------ ------  -----  ----- ------ ------ ------     ------- ------

  RESOLVED NUMBERS:                755     -     480    194     47     21     10      3     -      -      -      -         1161    1.54
                                   2.7%         13.1%   9.2%   5.0%   6.5%   9.2%   8.6%                                    9.1%
    COMPLETES                      21      -      -      1      7      8      5      -      -      -      -      -          80    3.81
                                   0.1%                  *    0.7%   2.5%   4.6%                                            0.6%
    OTHER RESOLVED                 734     -     480    193     40     13      5      3     -      -      -      -         1081    1.47
                                   2.6%         13.1%   9.2%   4.3%   4.0%   4.6%   8.6%                                    8.4%

  AVAILABLE NUMBERS:               780     -     248    294    177     46     13      2     -      -      -      -         1628    2.09
                                   2.8%          6.8%  14.0%  18.9%  14.2%  11.9%   5.7%                                   12.7%
    SYSTEM SCHEDULED:              476     -     243    152     63     16      2      -     -      -      -      -          810    1.70
                                   1.7%          6.7%   7.2%   6.7%   5.0%   1.8%                                           6.3%
      TODAY (BUCKET 0-7)           471     -     243    152     59     15      2      -     -      -      -      -          794    1.69
                                   1.7%          6.7%   7.2%   6.3%   4.6%   1.8%                                           6.2%
    TIMED TODAY                    215     -      2     106     77     21      9      -     -      -      -      -          574    2.67
                                   0.8%          0.1%   5.0%   8.2%   6.5%   8.3%                                           4.5%
    TIMED LATER                    54      -      -      23     25      4      1      1     -      -      -      -          148    2.74
                                   0.2%                 1.1%   2.7%   1.2%   0.9%   2.9%                                    1.2%
    SPCL INTV/HIDDEN               35      -      3      13     12      5      1      1     -      -      -      -           96    2.74
                                   0.1%          0.1%   0.6%   1.3%   1.5%   0.9%   2.9%                                    0.7%

TOTAL REPLICATE 03:               1534    28     835    414    185     53     15      4     -      -      -      -         2529    1.65
------------------                 5.5%   0.1%  22.9%  19.7%  19.7%  16.4%  13.8%  11.4%                                   19.7%

  RESOLVED NUMBERS:                697     -     476    134     59     20      7      1     -      -      -      -         1042    1.49
                                   2.5%         13.0%   6.4%   6.3%   6.2%   6.4%   2.9%                                    8.1%
    COMPLETES                      16      -      -      -      12      3      1      -     -      -      -      -           53    3.31
                                   0.1%                        1.3%   0.9%   0.9%                                           0.4%
    OTHER RESOLVED                 681     -     476    134     47     17      6      1     -      -      -      -          989    1.45
                                   2.4%         13.0%   6.4%   5.0%   5.3%   5.5%   2.9%                                    7.7%

  AVAILABLE NUMBERS:               837    28     359    280    126     33      8      3     -      -      -      -         1487    1.78
                                   3.0%   0.1%   9.8%  13.3%  13.4%  10.2%   7.3%   8.6%                                   11.6%
    SYSTEM SCHEDULED:              470    28     263    119     45     14      1      -     -      -      -      -          697    1.48
                                   1.7%   0.1%   7.2%   5.7%   4.8%   4.3%   0.9%                                           5.4%
      TODAY (BUCKET 0-7)           460    28     263    119     40     10      -      -     -      -      -      -          661    1.44
                                   1.6%   0.1%   7.2%   5.7%   4.3%   3.1%                                                  5.2%
    TIMED TODAY                    236     -      58    116     45     11      4      2     -      -      -      -          501    2.12
                                   0.8%          1.6%   5.5%   4.8%   3.4%   3.7%   5.7%                                    3.9%
    TIMED LATER                    92      -      36     33     15      6      1      1     -      -      -      -          182    1.98
                                   0.3%          1.0%   1.6%   1.6%   1.9%   0.9%   2.9%                                    1.4%
    SPCL INTV/HIDDEN               39      -      2      12     21      2      2      -     -      -      -      -          107    2.74
                                   0.1%          0.1%   0.6%   2.2%   0.6%   1.8%                                           0.8%

TOTAL REPLICATE 04:               1532   482     569    329    110     33      7      1      1     -      -      -         1737    1.13
------------------                 5.4%   2.3%  15.6%  15.7%  11.7%  10.2%   6.4%   2.9%  10.0%                            13.6%

  RESOLVED NUMBERS:                461     -     331     98     23      7      1      -      1     -      -      -          636    1.38
                                   1.6%          9.1%   4.7%   2.5%   2.2%   0.9%         10.0%                             5.0%
```

```
(continued)
                                   REPORT 10:  NUMBER OF CALLS MADE BY AVAILABILITY WITHIN REPLICATE

                                         NOT   <----------------------  NUMBER OF CALLS MADE  ---------------------->          AVERAGE
                               TOTAL   YET                                                               10 OR     TOTAL # CALLS
REPLICATE AVAILABILITY        NUMBERS CALLED   ONE    TWO    THREE  FOUR   FIVE    SIX    SEVEN  EIGHT  NINE   MORE      CALLS   MADE
----------------------        ------- ------   -----  -----  ----- ------ ------  -----  ----- ------ ------  ------   ------- ------

    COMPLETES                     10     -       -      3      4      2      1       -      -      -      -       -         31    3.10
                                   *                  0.1%   0.4%   0.6%   0.9%                                           0.2%
    OTHER RESOLVED               451     -     331     95     19      5      -       -      1      -      -       -        605    1.34
                                 1.6%          9.1%   4.5%   2.0%   1.5%                10.0%                             4.7%

  AVAILABLE NUMBERS:            1071   482     238    231     87     26      6       1      -      -      -       -       1101    1.03
                                 3.8%   2.3%   6.5%  11.0%   9.3%   8.0%   5.5%   2.9%                                    8.6%
    SYSTEM SCHEDULED:            808   480     168    120     32      4      3       1      -      -      -       -        541    0.67
                                 2.9%   2.3%   4.6%   5.7%   3.4%   1.2%   2.8%   2.9%                                    4.2%
      TODAY (BUCKET 0-7)         804   480     168    120     30      4      1       1      -      -      -       -        525    0.65
                                 2.9%   2.3%   4.6%   5.7%   3.2%   1.2%   0.9%   2.9%                                    4.1%
    TIMED TODAY                  168     -      45     72     33     15      3       -      -      -      -       -        363    2.16
                                 0.6%          1.2%   3.4%   3.5%   4.6%   2.8%                                           2.8%
    TIMED LATER                   79     2      25     32     16      4      -       -      -      -      -       -        153    1.94
                                 0.3%    *     0.7%   1.5%   1.7%   1.2%                                                  1.2%
    SPCL INTV/HIDDEN              16     -       -      7      6      3      -       -      -      -      -       -         44    2.75
                                 0.1%                 0.3%   0.6%   0.9%                                                  0.3%

TOTAL REPLICATE 05:             1534   774     525    169     47     15      3       1      -      -      -       -       1085    0.71
-------------------              5.5%   3.7%  14.4%   8.0%   5.0%   4.6%   2.8%   2.9%                                    8.5%

  RESOLVED NUMBERS:              307     -     230     57     14      5      -       1      -      -      -       -        412    1.34
                                 1.1%          6.3%   2.7%   1.5%   1.5%          2.9%                                    3.2%
    COMPLETES                      6     -       -      2      2      1      -       1      -      -      -       -         20    3.33
                                   *                  0.1%   0.2%   0.3%          2.9%                                    0.2%
    OTHER RESOLVED               301     -     230     55     12      4      -       -      -      -      -       -        392    1.30
                                 1.1%          6.3%   2.6%   1.3%   1.2%                                                  3.1%

  AVAILABLE NUMBERS:            1227   774     295    112     33     10      3       -      -      -      -       -        673    0.55
                                 4.4%   3.7%   8.1%   5.3%   3.5%   3.1%   2.8%                                           5.3%
    SYSTEM SCHEDULED:            973   774     128     50     16      3      2       -      -      -      -       -        298    0.31
                                 3.5%   3.7%   3.5%   2.4%   1.7%   0.9%   1.8%                                           2.3%
      TODAY (BUCKET 0-7)         968   774     128     50     15      -      1       -      -      -      -       -        278    0.29
                                 3.4%   3.7%   3.5%   2.4%   1.6%          0.9%                                           2.2%
    TIMED TODAY                  132     -      88     28      9      6      1       -      -      -      -       -        200    1.52
                                 0.5%          2.4%   1.3%   1.0%   1.9%   0.9%                                           1.6%
    TIMED LATER                  109     -      76     27      5      1      -       -      -      -      -       -        149    1.37
                                 0.4%          2.1%   1.3%   0.5%   0.3%                                                  1.2%
    SPCL INTV/HIDDEN              13     -       3      7      3      -      -       -      -      -      -       -         26    2.00
                                   *           0.1%   0.3%   0.3%                                                         0.2%

TOTAL REPLICATE 06:             1533  1270     188     56     16      1      2       -      -      -      -       -        362    0.24
-------------------              5.4%   6.1%   5.1%   2.7%   1.7%   0.3%   1.8%                                           2.8%

  RESOLVED NUMBERS:              101     -      82     14      5      -      -       -      -      -      -       -        125    1.24
                                 0.4%          2.2%   0.7%   0.5%                                                         1.0%
    COMPLETES                      1     -       -      -      1      -      -       -      -      -      -       -          3    3.00
                                   *                         0.1%                                                           *
```

```
                        Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 38

(continued)
                                  REPORT 10:  NUMBER OF CALLS MADE BY AVAILABILITY WITHIN REPLICATE

                                       NOT   <---------------------  NUMBER OF CALLS MADE  ---------------------->          AVERAGE
                              TOTAL   YET                                                                  10 OR     TOTAL # CALLS
REPLICATE AVAILABILITY       NUMBERS CALLED   ONE    TWO   THREE  FOUR   FIVE    SIX   SEVEN  EIGHT NINE   MORE      CALLS   MADE
---------------------        ------- ------  -----  -----  ----- ------ ------  -----  ----- ------ ------ ------   -------  ------

    OTHER RESOLVED            100        -    82     14        4      -      -      -      -      -      -      -       122    1.22
                                0.4%          2.2%   0.7%   0.4%                                                       1.0%

  AVAILABLE NUMBERS:          1432   1270    106     42     11      1      2      -      -      -      -      -       237    0.17
                                 5.1%  6.1%   2.9%   2.0%   1.2%   0.3%   1.8%                                         1.8%
    SYSTEM SCHEDULED:         1343   1270     48     18        6      1      -      -      -      -      -      -       106    0.08
                                 4.8%  6.1%   1.3%   0.9%   0.6%   0.3%                                                0.8%
      TODAY (BUCKET 0-7)      1343   1270     48     18        6      1      -      -      -      -      -      -       106    0.08
                                 4.8%  6.1%   1.3%   0.9%   0.6%   0.3%                                                0.8%
    TIMED TODAY                 56       -    38     14        4      -      -      -      -      -      -      -        78    1.39
                                0.2%          1.0%   0.7%   0.4%                                                       0.6%
    TIMED LATER                 27       -    19       6       1      -      1      -      -      -      -      -        39    1.44
                                0.1%          0.5%   0.3%   0.1%          0.9%                                         0.3%
    SPCL INTV/HIDDEN              6      -      1      4      -      -      1      -      -      -      -      -        14    2.33
                                  *             *    0.2%                 0.9%                                         0.1%

TOTAL REPLICATE 07:           1534   1526      8      -      -      -      -      -      -      -      -      -         8    0.01
-------------------             5.5%   7.3%  0.2%                                                                      0.1%

  RESOLVED NUMBERS:               1      -      1      -      -      -      -      -      -      -      -      -         1    1.00
                                  *             *                                                                        *
    COMPLETES                     1      -      1      -      -      -      -      -      -      -      -      -         1    1.00
                                  *             *                                                                        *

  AVAILABLE NUMBERS:          1533   1526      7      -      -      -      -      -      -      -      -      -         7    0.00
                                 5.4%  7.3%   0.2%                                                                     0.1%
    SYSTEM SCHEDULED:         1529   1526       3      -      -      -      -      -      -      -      -      -        3    0.00
                                 5.4%  7.3%   0.1%                                                                       *
      TODAY (BUCKET 0-7)      1529   1526       3      -      -      -      -      -      -      -      -      -        3    0.00
                                 5.4%  7.3%   0.1%                                                                       *
    TIMED TODAY                   2      -      2      -      -      -      -      -      -      -      -      -         2    1.00
                                  *           0.1%                                                                       *
    SPCL INTV/HIDDEN              2      -      2      -      -      -      -      -      -      -      -      -         2    1.00
                                  *           0.1%                                                                       *

TOTAL REPLICATE 08:           1533   1530      2      -      -      -      -      1      -      -      -      -         8    0.01
-------------------             5.4%   7.3%  0.1%                                2.9%                                  0.1%

  RESOLVED NUMBERS:               1      -      -      -      -      -      -      1      -      -      -      -         6    6.00
                                  *                                              2.9%                                    *
    OTHER RESOLVED                1      -      -      -      -      -      -      1      -      -      -      -         6    6.00
                                  *                                              2.9%                                    *

  AVAILABLE NUMBERS:          1532   1530      2      -      -      -      -      -      -      -      -      -         2    0.00
                                 5.4%  7.3%   0.1%                                                                       *
    SYSTEM SCHEDULED:         1530   1530       -      -      -      -      -      -      -      -      -      -        -    0.00
                                 5.4%  7.3%
      TODAY (BUCKET 0-7)      1530   1530       -      -      -      -      -      -      -      -      -      -        -    0.00
                                 5.4%  7.3%
```

```
(continued)
                                REPORT 10:  NUMBER OF CALLS MADE BY AVAILABILITY WITHIN REPLICATE

                                         NOT   <---------------------  NUMBER OF CALLS MADE   ---------------------->          AVERAGE
                               TOTAL   YET                                                                     10 OR     TOTAL # CALLS
REPLICATE AVAILABILITY        NUMBERS CALLED   ONE    TWO   THREE  FOUR   FIVE     SIX   SEVEN  EIGHT  NINE   MORE      CALLS   MADE
----------------------         ------- ------  -----  -----  ----- ------ ------  -----  ----- ------ ------ ------    ------- ------

    SPCL INTV/HIDDEN               2      -      2       -      -      -      -      -      -      -      -      -           2    1.00
                                   *           0.1%                                                                          *

TOTAL REPLICATE 09:             1534   1528      4       2      -      -      -      -      -      -      -      -           8    0.01
-------------------              5.5%   7.3%   0.1%   0.1%                                                                 0.1%

  RESOLVED NUMBERS:                2      -      -       2      -      -      -      -      -      -      -      -           4    2.00
                                   *                  0.1%                                                                   *
    COMPLETES                      2      -      -       2      -      -      -      -      -      -      -      -           4    2.00
                                   *                  0.1%                                                                   *

  AVAILABLE NUMBERS:            1532   1528      4       -      -      -      -      -      -      -      -      -           4    0.00
                                 5.4%   7.3%   0.1%                                                                          *
    SYSTEM SCHEDULED:           1530   1528      2       -      -      -      -      -      -      -      -      -           2    0.00
                                 5.4%   7.3%   0.1%                                                                          *
      TODAY (BUCKET 0-7)        1530   1528      2       -      -      -      -      -      -      -      -      -           2    0.00
                                 5.4%   7.3%   0.1%                                                                          *
    SPCL INTV/HIDDEN               2      -      2       -      -      -      -      -      -      -      -      -           2    1.00
                                   *           0.1%                                                                          *

TOTAL REPLICATE 10:             1533   1529      1       3      -      -      -      -      -      -      -      -           7    0.00
-------------------              5.4%   7.3%     *    0.1%                                                                 0.1%

  RESOLVED NUMBERS:                3      -      -       3      -      -      -      -      -      -      -      -           6    2.00
                                   *                  0.1%                                                                   *
    COMPLETES                      1      -      -       1      -      -      -      -      -      -      -      -           2    2.00
                                   *                     *                                                                   *
    OTHER RESOLVED                 2      -      -       2      -      -      -      -      -      -      -      -           4    2.00
                                   *                  0.1%                                                                   *

  AVAILABLE NUMBERS:            1530   1529      1       -      -      -      -      -      -      -      -      -           1    0.00
                                 5.4%   7.3%     *                                                                           *
    SYSTEM SCHEDULED:           1530   1529      1       -      -      -      -      -      -      -      -      -           1    0.00
                                 5.4%   7.3%     *                                                                           *
      TODAY (BUCKET 0-7)        1530   1529      1       -      -      -      -      -      -      -      -      -           1    0.00
                                 5.4%   7.3%     *                                                                           *

TOTAL REPLICATE 11:             1532   1531      1       -      -      -      -      -      -      -      -      -           1    0.00
-------------------              5.4%   7.3%     *                                                                           *

  AVAILABLE NUMBERS:            1532   1531      1       -      -      -      -      -      -      -      -      -           1    0.00
                                 5.4%   7.3%     *                                                                           *
    SYSTEM SCHEDULED:           1532   1531      1       -      -      -      -      -      -      -      -      -           1    0.00
                                 5.4%   7.3%     *                                                                           *
      TODAY (BUCKET 0-7)        1532   1531      1       -      -      -      -      -      -      -      -      -           1    0.00
                                 5.4%   7.3%     *                                                                           *

TOTAL REPLICATE 12:             1535   1532      2       1      -      -      -      -      -      -      -      -           4    0.00
-------------------              5.5%   7.3%   0.1%      *                                                                   *
```

```
(continued)
                                              REPORT 10:  NUMBER OF CALLS MADE BY AVAILABILITY WITHIN REPLICATE

                                          NOT    <---------------------  NUMBER OF CALLS MADE  ---------------------->          AVERAGE
                                  TOTAL   YET                                                                 10 OR      TOTAL # CALLS
REPLICATE AVAILABILITY          NUMBERS CALLED   ONE    TWO   THREE  FOUR   FIVE    SIX  SEVEN  EIGHT  NINE   MORE       CALLS   MADE
----------------------          ------- ------   -----  -----  ----- ------ ------  ----- ----- ------ ------ ------     ------- ------

  AVAILABLE NUMBERS:             1535   1532      2       1      -       -      -      -      -      -       -      -         4    0.00
                                  5.5%   7.3%   0.1%      *                                                                  *
    SYSTEM SCHEDULED:            1533   1532      1       -      -       -      -      -      -      -       -      -         1    0.00
                                  5.4%   7.3%     *                                                                          *
      TODAY (BUCKET 0-7)         1533   1532      1       -      -       -      -      -      -      -       -      -         1    0.00
                                  5.4%   7.3%     *                                                                          *
    TIMED TODAY                     1      -      -       1      -       -      -      -      -      -       -      -         2    2.00
                                    *                     *                                                              *
    SPCL INTV/HIDDEN                1      -      1       -      -       -      -      -      -      -       -      -         1    1.00
                                    *             *                                                                          *

TOTAL REPLICATE 13:              1535   1533      1       1      -       -      -      -      -      -       -      -         3    0.00
-------------------               5.5%   7.3%     *       *                                                                  *

  AVAILABLE NUMBERS:             1535   1533      1       1      -       -      -      -      -      -       -      -         3    0.00
                                  5.5%   7.3%     *       *                                                                  *
    SYSTEM SCHEDULED:            1535   1533      1       1      -       -      -      -      -      -       -      -         3    0.00
                                  5.5%   7.3%     *       *                                                                  *
      TODAY (BUCKET 0-7)         1535   1533      1       1      -       -      -      -      -      -       -      -         3    0.00
                                  5.5%   7.3%     *       *                                                                  *

TOTAL REPLICATE 14:              1535   1535      -       -      -       -      -      -      -      -       -      -         -    0.00
-------------------               5.5%   7.3%

  AVAILABLE NUMBERS:             1535   1535      -       -      -       -      -      -      -      -       -      -         -    0.00
                                  5.5%   7.3%
    SYSTEM SCHEDULED:            1535   1535      -       -      -       -      -      -      -      -       -      -         -    0.00
                                  5.5%   7.3%
      TODAY (BUCKET 0-7)         1535   1535      -       -      -       -      -      -      -      -       -      -         -    0.00
                                  5.5%   7.3%

TOTAL REPLICATE 15:              1535   1529      3       1      1       -      1      -      -      -       -      -        13    0.01
-------------------               5.5%   7.3%   0.1%      *    0.1%          0.9%                                          0.1%

  AVAILABLE NUMBERS:             1535   1529      3       1      1       -      1      -      -      -       -      -        13    0.01
                                  5.5%   7.3%   0.1%      *    0.1%          0.9%                                          0.1%
    SYSTEM SCHEDULED:            1534   1529      2       1      1       -      1      -      -      -       -      -        12    0.01
                                  5.5%   7.3%   0.1%      *    0.1%          0.9%                                          0.1%
      TODAY (BUCKET 0-7)         1533   1529      2       1      1       -      -      -      -      -       -      -         7    0.00
                                  5.4%   7.3%   0.1%      *    0.1%                                                        0.1%
    SPCL INTV/HIDDEN                1      -      1       -      -       -      -      -      -      -       -      -         1    1.00
                                    *             *                                                                          *

TOTAL REPLICATE 16:              1534   1534      -       -      -       -      -      -      -      -       -      -         -    0.00
-------------------               5.5%   7.3%

  AVAILABLE NUMBERS:             1534   1534      -       -      -       -      -      -      -      -       -      -         -    0.00
                                  5.5%   7.3%
```

```
                               Computers for Marketing - Phone report for eid572.fon -  1:23 PM 13 FEB 2001 - Page 41

(continued)
                                        REPORT 10:  NUMBER OF CALLS MADE BY AVAILABILITY WITHIN REPLICATE

                                          NOT   <---------------------  NUMBER OF CALLS MADE  ---------------------->              AVERAGE
                                  TOTAL  YET                                                                  10 OR      TOTAL # CALLS
REPLICATE AVAILABILITY           NUMBERS CALLED   ONE    TWO   THREE  FOUR   FIVE    SIX   SEVEN  EIGHT  NINE   MORE      CALLS   MADE
---------------------            ------- ------  -----  -----  ----- ------ ------  -----  ----- ------ ------ ------    ------- ------

    SYSTEM SCHEDULED:               1534   1534     -      -      -      -      -      -      -      -      -      -          -    0.00
                                     5.5%   7.3%
      TODAY (BUCKET 0-7)            1534   1534     -      -      -      -      -      -      -      -      -      -          -    0.00
                                     5.5%   7.3%

TOTAL REPLICATE 17:                 1533   1530     -      1      2      -      -      -      -      -      -      -          8    0.01
-------------------                  5.4%   7.3%           *    0.2%                                                      0.1%

  RESOLVED NUMBERS:                    2      -     -      1      1      -      -      -      -      -      -      -          5    2.50
                                       *                   *    0.1%                                                         *
    COMPLETES                          1      -     -      -      1      -      -      -      -      -      -      -          3    3.00
                                       *                        0.1%                                                         *
    OTHER RESOLVED                     1      -     -      1      -      -      -      -      -      -      -      -          2    2.00
                                       *                   *                                                                 *

  AVAILABLE NUMBERS:                1531   1530     -      -      1      -      -      -      -      -      -      -          3    0.00
                                     5.4%   7.3%                0.1%                                                         *
    SYSTEM SCHEDULED:               1531   1530     -      -      1      -      -      -      -      -      -      -          3    0.00
                                     5.4%   7.3%                0.1%                                                         *
      TODAY (BUCKET 0-7)            1531   1530     -      -      1      -      -      -      -      -      -      -          3    0.00
                                     5.4%   7.3%                0.1%                                                         *

TOTAL REPLICATE 18:                 1532   1530     1      1      -      -      -      -      -      -      -      -          3    0.00
-------------------                  5.4%   7.3%    *      *                                                                 *

  RESOLVED NUMBERS:                    1      -     -      1      -      -      -      -      -      -      -      -          2    2.00
                                       *                   *                                                                 *
    COMPLETES                          1      -     -      1      -      -      -      -      -      -      -      -          2    2.00
                                       *                   *                                                                 *

  AVAILABLE NUMBERS:                1531   1530     1      -      -      -      -      -      -      -      -      -          1    0.00
                                     5.4%   7.3%    *                                                                        *
    SYSTEM SCHEDULED:               1530   1530     -      -      -      -      -      -      -      -      -      -          -    0.00
                                     5.4%   7.3%
      TODAY (BUCKET 0-7)            1530   1530     -      -      -      -      -      -      -      -      -      -          -    0.00
                                     5.4%   7.3%
    SPCL INTV/HIDDEN                   1      -     1      -      -      -      -      -      -      -      -      -          1    1.00
                                       *            *                                                                        *




Note: Percentage less than 0.05 printed as *.
```

APPENDIX A: SAMPLE SPECIFICATIONS

1. BANK

This matches the HP3000-SPL sample bank questionnaire, except that question numbers are replaced by labels when needed. And !CAT questions are replaced by !FLD questions. Required syntax and interviewer notes are in UPPER-CASE; nonrequired syntax, question labels and text to be spoken are not in upper-case text.

Question numbers or specific data locations would not be used in practical applications.

Directs program to **compile**	~PREPARE COMPILE
Study header: NEWBNK (study name), length =7/40. ID on every record in 1.4, card ID in 80, interviewers can abort, text data starts in 6/1, user-assigned columns starting at 2/15	[NEWBNK, 7/40, “Bank demo questionnaire”, & SPEC_WIDTH=132,CARD_FORMAT= 80 & TEXT_START=6/1,WORK_START=2/15]
	{!ALLOW_ABORT}
{!AUTO_PUNCHES} makes the response code be the stored punch for CAT questions	{!AUTO_PUNCHES}

\((5.5) sets a permanent box for the questions at column 5, row 5 of the screen (every question will be indented like this

```
{
\((5.5)

   [NOTE: PRIOR TO THE INTERVIEW,
   RESPONDENTS HAVE BEEN SCREENED
   TO INCLUDE ONLY INDIVIDUALS, AGE 18
   AND OVER, WHO ARE HEAD OF
   HOUSEHOLD. QUOTAS BASED ON
   NUMBER OF MALES AND FEMALES HAVE
   BEEN MET]

Hello, I'm ... of Financial Decisionmakers, an
independent opinion research firm. We would
like to ask you opinions on banking services and
financial investments.

1. Do you or any member of your household
have credit cards?

!FLD
1 YES
(SKIPTO Services) 2 NO
(SKIPTO Services) Y DK/NA }

{ Cardtype

2A. Which of the following types of credit cards
to you, or any household members have?

          *** READ THE NAME OF EACH TYPE
          AND ENTER ALL THAT APPLY ***

2A. Which of the following types of credit cards
to you, or any household members have?

!FLD,,6
1 GENERAL PURPOSE CARDS
2 BANK CARDS
3 RETAIL STORE
4 GAS/OIL COMPANY CARDS
5 CAR RENTAL/AIRLINE
6 OTHER
```

{!SAVE_COLUMN} saves the current column as 'A', so that the next question can be placed at the end and have the program return here for subsequent questions

```
{!SAVE_COLUMN A}

{ Othrcard: 2/13
!IF Cardtype (6)
What other type of credit card is that?

          *** ENTER TYPE OF CREDIT CARD ***

!TEX}
```

{!RESTORE_COLUMN} restores the column to 'A'

```
{!RESTORE_COLUMN A}
```

>REPEAT repeats the specifications to create questions for each of the possible credit cards used

NOTE:

The ellipsis (...) can be used to continue an increment.

Strings need to be in quotes if there are blanks or special characters.

The variable names ($A, $B) can be used anywhere in the spec, and as often as you like.

```
>REPEAT $A=1,...,6;$B="General Purpose", &
"Car Rental/Airline",\:Othrcard:"

{
!IF Cardtype($A)
2B. How often do you use your \B$B\Ecards - at
least once a month, at least every 3 months, a
few times each year, less than that, or never?

!FLD
1 AT LEAST MONTHLY
2 EVERY 3 MONTHS
3 A FEW TIMES PER YEAR
4 LESS
5 NEVER}
```

>END_REPEAT ends this repeat block

```
>END_REPEAT
```

This **>**starts another repeat block. The repeat names can be up to 16 characters.

```
>REPEAT $PUN=1,...,6;$COL=07,...,12

{
!IF [1/6^N$PUN]
!GEN,A,[$COL],6}
{
```

>END_REPEAT is the end of this repeat block.

```
>END_REPEAT

(Services:
3A. The following is a list of banking services. Please
tell me which services you currently use.

      *** READ THE NAME OF EACH TYPE AND
      ENTER ALL THAT APPLY ***

!FLD,,6
1 CHECKING ACCOUNT
2 SAVINGS ACCOUNT
3 SAVINGS CERTIFICATES
4 MORTGAGE
5 BANK CARDS
6 OTHER LOANS
(-) 0 (None of these)}
```

If you choose 0, no answer may be chosen (-).

This repeat block includes data locations, question labels and titles.

```
>REPEAT $A=1,...,6;$B=“CHECKING ACCOUNT”,&
“SAVINGS ACCOUNT”,”SAVINGS CERTIFICATES”,&
“MORTGAGE”,”BANK CARDS”,”OTHER LOANS”;&
$C=”2/43”,”3/5”,”3/40”,”4/5”,”4/40”,”5/5”
```

Notice the use of **bold-facing** of the repeated titles (\B).

```
{Acct$A: $C
!IF Services ($A)
3B. Which financial institution holds your largest
account for your

\B$B\E ?
***ENTER NAME OF FINANCIAL INSTITUTION***

!VAR..35,1}
>END_REPEAT
```

You can use the same repeat names ($A,$B) over again in separate repeats and assign them different values each time.

This uses bolding and back-referencing of the responses to the Acct# questions.

```
>REPEAT $A=1,...,6;$B="CHECKING ACCOUNT",&
"SAVINGS ACCOUNT","SAVINGS CERTIFICATES",&
"BANK CARDS","OTHER LOANS"

{ #3.7$A
!IF Services ($A)
3C. How would you rate the quality of services
received at

\B\:Acct$A:\E for your \B$B\E -
is it excellent, good, fair or poor?|

!FLD
1 EXCELLENT
2 GOOD
3 FAIR
4 POOR
Y DK/NA }

>END_REPEAT

{Ownrent:
4. Do you currently own or rent your home?

!FLD
1 OWN
2 RENT
Y DK/NA }

{HHcost
!IF Ownrent(1)
5. If you sold your home today, approximately
how much would you receive for it?

*** ENTER THE DOLLAR AMOUNT
(>=10,000 ***
or DK for Don't know

!NUM,,,10000-9999999,,DK}
```

```
{Invest:
6A. Now I'm going to read a list of types of
investments.

Please tell me if you currently have this type of
investment.
*** READ THE NAME OF EACH TYPE AND
ENTER ALL THAT APPLY ***

!FLD,,7
1 RENTAL PROPERTIES
2 NON INCOME-PRODUCING PROPERTIES
3 MARKETABLE GOODS (i.e. gold, art)
4 STOCKS/BONDS
5 TAX SHELTERS
6 MONEY-MARKET CERTIFICATES
7 SAVINGS ACCOUNTS
(-) Y (None of these) }

>REPEAT $A=1,...,7;$B="RENTAL PROPERTIES",&
"NON INCOME-PRODUCING PROPERTIES",&
"MARKETABLE GOODS","STOCKS/BONDS"&
"TAX SHELTERS",&
"MONEY-MARKET CERTIFICATES",&
"SAVINGS ACCOUNTS"

{IF Invest ($A)
6B. What is the approximate dollar value of your

\B$B\E ?

!NUM,SAMEAS HHCOST}

>EBD_REPEAT

{Media: 1/78
7. From which sources do you obtain information
for your personal financial and investment
plans?

*** DO NOT READ CATEGORIES
ENTER ALL TYPES MENTIONED ***
```

```
!FLD,,11
1 TV NEWS
2 TV SPECIAL BUSINESS PROGRAMS
3 LOCAL DAILY NEWSPAPERS
4 SPECIAL BUSINESS NEWSPAPERS
5 SUNDAY BUSINESS SECTION
6 BUSINESS MAGAZINES
7 NEWS MAGAZINES
8 BOOKS
9 FINANCIAL PROFESSIONALS
0 BANK LEAFLETS
(-) Y (None of these sources) }

{!COLUMN 2/5}

{Occup:
Now, just a few questions for backround
purposes ...

8. What is your occupation?

*** DO NOT READ CATEGORIES.
ENTER THE APPROPRIATE CODE ***

!FLD
1 PROFESSIONAL/TECHNICAL
2 MANAGER/ADMINISTRATOR
3 SALES WORKER
4 CLERICAL WORER
5 CRAFTS WORKER
6 SERVIE WORKER
7 LABORER
8 GOVERNMENT/MILITARY
9 OTHER
0 UNEMPLOYED/RETIRED
Y N/A}

{Educ:
9. What is the highest level of formal education
completed?
```

```
!FLD
1 DID NOT GRADUATE HIGH SCHOOL
2 GRADUATED HIGH SCHOOL
3 TECHNICAL SCHOOL
4 ATTENDED COLLEGE
5 GRADUATED COLLEGE (4 YEARS)
6 POST GRADUATE WORK
Y N/A}

{Income:
10. Which of the following categories
approximates your total household yearly
income?

!FLD
1 UNDER $7,500
2 $7,500-$9,999
3 $10,000-$14,999
4 $15,000-$19,999
5 $20,000-$24,999
6 $25,000-$29,999
7 $30,000-$39,999
8 $40,000-$49,999
9 $50,000-$74,999
0 $75,000-$99,999
X $100,000 AND OVER
Y N/A (do not read) }

{Married
11. What is your marital status?

*** ENTER APPROPRIATE LETTER CODE ***

!FLD
S SINGLE
L LIVING TOGETHER, NOT MARRIED
M MARRIED
D SEPARATED/DIVORCED
W WIDOWED
K N/A }
```

```
{Hhsize:
12. Number of individuals in your household
(including you).

!NUM,,,1-15

{Age:
13. What is your age?

*** ENTER AGE CATEGORY ***

!FLD
1 18-24
2 25-29
3 30-34
4 35-44
5 55-54
7 65 and over \N
Y No Answer }

{Sex:

*** THANK THE RESPONDENT AND
TERMINATE THE INTERVIEW***

14. RECORD SEX OF RESPONDENT.

!FLD
1 MALE \N
2 FEMALE }

~END
```

2. ROADRUNR

This is the ROADRUNR questionnaire used throughout the MENTOR manual. Data locations are hard-coded to match the data file Mentor users have access to. Required syntax, question labels, and interviewer notes are in upper-case; nonrequired syntax and text to be spoken are not in all upper-case.

```
~PREPARE COMPILE
>PURGE_SAME
```

The area for **TEX answers** starts at 5/1 (column 321). There's a work area starting at 4/1 (column 241).

```
[RRUNR,640,"EXAMPLEJOB",&
SPEC_WIDTH=132,&
TEXT_START=5/1,WORK_START=4/1]
```

!{BLANK_LINES=1} adds a blank line between question text and the response list (if any) or prompt (if no response list).

```
{ !AUTO_PUNCHES }
{ !BLANK_LINES=1 }

{ QN: 1/6.1
Q1. How much do you agree with the
following statement:
The fast food at Road Runners is worth what
I pay for it.
```

5 is the response code, while (5) starts the question text.

```
!FLD
5 (5) Completely agree
4 (4) Somewhat agree
3 (3) Neither agree nor disagree
2 (2) Somewhat agree
1 (1) Completely disagree
0 Don't know/Refused to answer
}
```

Hard coding the width of the question (with .1) ensures a complaint from PREPARE if it thinks it needs more columns.

```
{ QN2A: 1/7.1
Q2a. Please rate the following
characteristics:
The quality of the food

!FLD
5 (5) Very good
4 (4) Good
3 (3) Neither poor nor good
2 (2) Poor
1 (1) Very poor
0 Don’t know/refused to answer

>REPEAT $A=B,...,F;$B=08,...,12;&
$C= “quality of service”,&
“cleanliness of restaurant”,&
“prices on the menu”,&
“accuracy of your bill”,&
“cleanliness of restrooms”

{ QN2$A: 1/$B.1
Q2$A. Please rate the following
characteristics:
The $C
```

[SAMEAS] says to use the same response list as a previous question.

```
!FLD
[SAMEAS QN2A]
}

>END_REPEAT
```

If you leave off the width (.1), the program will figure it out. Specify the width to make sure PREPARE doesn't assign a different number of columns than you wanted.

```
{ Q2NG: 1/13.1
Q2g. What is your opinion of the Road
Runners bill format?

Is it . . .
!FLD
1 Easy to understand
2 Hard to understand
3 Neither easy nor hard to understand
0 Don't know
}
```

```
{ QN3: 1/14.1
Q3. Now, in thinking about what your Road
Runners fast food costs - not your
entertainment charges - but the food, would
you say your fast food cost is . . .

!FLD
5 (5) A very good value
4 (4) A good value
3 (3) Neither a good nor a poor value
2 (2) A poor value
1 (1) A very poor value
0 Don't know/refused to answer
}

{ QN4: 1/15.2
Q4. About how much do ytou pay per visit for
Road Runners fast food - that is, not
including entertainment?

***INTERVIEWER: ENTER WHOLE
NUMBER (5-50) OR "RF" FOR REFUSED
***

!NUM,,,5-50,,RF
}
```

Allowable numbers are numbers from 5-50 or “RF”. It’s a good idea to tell the interviewer what the exception codes are.

Remember **!AUTO_PUNCHES** will assign punches according to the response code.

```
{QN5A: 1/20.1
Q5a. Rate the following charges at Road
Runners:
The basic rate for fast food.
!FLD
5 (5) Very reasonable
4 (4) Reasonable
3 (3) Neither reasonable nor unreasonable
2 (2) Not very reasonable
1 (1) Unreasonable charges
0 Don’t know/refused to answer


>REPEAT $A=B,...,F;$B=21,...,25;&
$C=”side orders”,&
“daily specials“,&
“drinks”,&
“optional services such as entertainment”,&
“vegetarian specials”
```

The text "The change for" was part of every statement, so it was kept out of the repeat text. Only try to include things that are changing.

```
{ QN5$A: 1/B.1
Q5$A. Rate the following charges at Road
Runners:
The charge for $C.
!FLD
[SAMEAS QN5A]
}
>END_REPEAT

{ QN5G: 1/26.1
Q5g. In response to the statement, "Road
Runners fast food provides good food at
realistic and affordable rates."
Do you . . .
!FLD
[SAMEAS QN1]
}

{ QN6: 1/30.1
Q6. Has anyone in your household
participated in the entertainment available at
Road Runners during the past three months?
!FLD
1 YES
2 NO
0 Don't Know
```

This question will only be asked of people who said YES to the previous question.

```
{ QN^A: 1/31.1
!IF QN6(1)
Q6a. Which entertainment was participated
in during the past three months?
!FLD,,6
1 Video games
2 Billiards
3 Fun House
4 Musican Revue
5 Dunk the Moose
6 Other
```

This dash (-) indicates a single response code

```
(-) Don’t know/refused
}

{ QN6b: 1/32.1
!IF QN6 (1) and not (QN6A (1))
Q6b. Have you used the video games at
Road Runners within the past three months?
!FLD
[SAMEAS QN6]
}
```

Sometimes you have to **split long lines of text** into two $variables to make it fit on your screen (in your box).

```
>REPEAT $A=A,...,G;$B=33,...,39,&
$C=”entertainment uses the most
advanced”,&
“is a fun place to go”,&
“does a good job of keeping customers”,&
“makes it easy for people to relax”,&
“has fast and efficient service”,&
“is a good plae to bring the kids”,&
“is dedicated to excellence”;&
$D=”technology”,” ”,”happy”,” ”,” ”,” ”,” ”

QN7$A: 1/$B.1
Q7$A. How do you feel about the following
statements:
Road Runners $C
$D.
!FLD
[SAMEAS QN1]
}
>END_REPEAT
```

!FLD,,15 allows up to 15 responses.

(-) makes these reponses **exclusive**. You can't pick them in combination with another response.

```
{ QN8: 1/40.2
Q8. From your own experience and
knowledge, what do you especially like about
Road Runners?
!FLD,,15
01 Good service/prompt service
02 Dependable/continuous service
03 The courteous employees they
have/helpful
04 I like the food/good food
05 Food selection
06 Good prices
07 Computerized/accurate billing
08 Helpful in explaining billing questions
09 Good entertainment
10 Variety of entertainment
11 Nice family atmosphere
12 Established place/been around for awhile
13 Accessible/Available/They're everywhere
14 Like everything/good place
15 Other
(-) 22 No problem/No complaints
(-) 23 Don't know/No answer
(-) 24 Nothing
}
```

```
{ QN9: 1/42.2
Q9. From your own experience and
knowledge, what do you especially dislike
about Road Runners?
!FLD,,19
01 Lack of sensitivity to special orders
02 Substitutions not allowed
03 Slow/poor service
04 Difficult/discourteous employees
05 High prices for side orders
06 High prices for specials
07 High prices (general)
08 Dirty restrooms
09 Not enough variety in food
10 Not enough variety in food
11 Not enough adult entertainment
12 Entertainment charges too high
13 Being charged for some entertainment
14 Extra charges for borthday party room
15 Inaccurate/incorrect billing
16 Inconvenient to get change for video
games
17 Dislike loud background music
18 Musical revue is not good
22 Other
(-) 23 Don’t Know/No answer
(-) 24 Nothing
}

{ QN10: 1/44.1
Q10. When you eat at fast food restaurants,
do you eat mostly at Road RUnners or
another restaurant?
!FLD
1. Road Runners
2 Another restaurant
0 Don’t know/Refused
}
```

Ask this question only if the response to question QN10 was 1.

```
{ QN10A: 1/45.1
!IF QN10 (1)
Q10a. Do you go mostly for the food or the
entertainment?
!FLD
1 Food
2 Entertainment
3 Both equally
0 Don’t know
}
```

Ask this question only if the response to question QN10 was 2.

```
{ QN11: 1/46.1
!IF QN10 (2)
Q11. What entertainment would you like to
see added at Road Runners?
!FLD,,10
01 Basketball hoop
02 Whack-a-Mole
03 More video games
04 Specific video game
05 Music videos
06 Darts
07 Foosball
08 More entertainment for teenagers
09 More entertainment for small children
10 Child care facilities
11 Other
(-) 12 Don’t know/Refused
}

{ QN12: 1/47.1
Q12. Overall, how would you rate Road
Runners?
!FLD
[SAMEAS QN2A]
}
```

```
{ QN13: 1/48.1
Q13. How long have you lived in your current
home?
!FLD
1 Less than one year
2 1 to 3 years
3 4 to 5 years
4 6 to 10 years
5 Longer than 10 years
0 Refused
}
```

These response codes could have been written as single-digit codes. It's usually better for the interviewer to be consistent when possible.

```
{ QN14: 1/49
Q14. What is the occupation of the principal
wage earner in your home?
!FLD
01 Professional/Technical
02 Clerical
03 Skilled laborer
04 Unskilled laborer
05 Housewife/Student
06 Unemployed
07 Retired
08 Other
09 Refused
}

{ QN15: 1/50.1
Q15. What is the highest level of formal
education completed by the principal wage
earner in your home?
!FLD
1 Less than grade school
2 Grade school
3 Some high school
4 High school diploma
5 Technical school
6 Some college
7 College graduate
8 Graduate school
9 Refused
}

{ QN16: 1/51.1
Q16. Which of the following best describes
your age?
!FLD
1 Under 25
2 25 to 34
3 35 tp 44
4 45 to 54
5 55 to 64
665 or over
7 Don't know/Refused
}
```

```
{ QN17: 1/52.1
Q17. Into which of the following categories
does your total family incomce fall?
!FLD
1 Under $15,000
2 $15,000 - $24,999
3 $25,000 - $34,999
4 $35,000 - $44,999
5 $45,000 - $49,999
6 $50,000 or more
9 Don't know/Refused
}

{ QN18: 1/53.1
Q18. Would you classify your ethnic
background as ...
!FLD
1 Caucasian
2 African-American
3 Hispanic
4 Oriental
5 American Indian
9 Some other ethnic group
0 Refused
}

{QN19: 1/54.1
Q19. And how would you describe your
marital status?
1 Married
2 Divorced
3 Widowed
4 Single, never married
5 Living together
6 Refused
}

{QN20: 1/55.2
Q20. And how many children ages 18 and
under are living at home?
!NUM,,,0-10,,RF
```

```
{QN21: 1/57.1
Q21/ ***INTERVIEWER: INDICATE SEX OF
RESPONDENT ***
!FLD
1 Male
2 Female
3 Not determined
}
```

Var,subtype L only allows alphabetic characters and blanks. No numbers or special characters are allowed. This won’t work if names are hyphenated.

```
{ DEMO1: 4/1
May I have your name so that my supervisor
can verify this information if necessary?
!VAR,L,35,5
}
```

Allow up to 10 lines of text.

```
{ DEMO2: 4/71
And are there any comments you wish to
make about this questionnaire or your visit to
Road Runners?
!TEX,,10
}
```

!FLD,subtype N won't display the response list on the screen.

```
{ LAST: 1/58.2
*** INTERVIEWER: ENTER TWO
CHARACTER STATE CODE FROM PAPER
SAMPLE ***
!FLD,N
AK Alaska
AL Alabama
AR Arkansas
AZ Arizona
CA California
CO Colorado
CT Connecticut
DC District of Columbia
DE Delaware
FI Florida
GA Georgia
HI Hawaii
ID Idaho
IL Illinois
IN Indiana
IA Iowa
KS Kansas
KY Kentucky
LA Louisianna
MA Massachusetts
MD Maryland
ME Maine
MI Michigan
MN Minnesota
MO Missouri
MS Mississippi
MT Montana
NC North Carolina
ND North Dakota
NE Nebraska
NH New Hampshire
NJ New Jersery
NM New Mexico
NV Nevada
NY New York
OH Ohio
OK Oklahoma
OR Oregon
PA Pennsylvania
RI Rhode Island
SC South Carolina
SD South Dakota
TN Tennessee
TX Texas
UT Utah
VA Virginia
VT Vermont
WA Washington
MI Wisconsin
WV West Virginia
WY Wyoming
}

~END
```

APPENDIX B: ALLOWED ABBREVIATIONS

B

Most keywords, commands, options and question types can be abbreviated. In most cases, the unabbreviated version is used throughout the manual. Once you become more familiar with how the different items work, you may want to use the abbreviated versions for ease of spec writing.

For more information on these commands, refer to the key to reference numbers, which will direct you to the appropriate section in the manual.

Key to reference numbers:

1 Study Header Option (2.3.2)

2 Interview Control (3.2.1)

3 Compos Control (3.2.2)

4 Compile Control (3.2.3)

5 Data Control (3.2.4)

6 Data Entry Question types (2.4)

7 Control Statements (3.1)

8 Composing and Compiling Questions in Prepare (2.3.1)

9 The Phone Statement (6.1.1)

10 Help Compiler Commands (3.2.5)

Item	Abbreviation	Reference
After_quit	AFTERQUIT	2
Allow_abort	ABORT	1
Allow_abort	ALLOWABORT	2
Allow-backup	ALLOWBACKUP	2
Allow_data_overlap	ALLOWOVERLAP	1
Allow_monitor	ALLOWMON	2

Item	Abbreviation	Reference
Allow_reset	ALLOWRSET	2
Allow_retake	ALLOWRETAKE	2
Allow_special	ALLOWPSPECIAL	2
Allow_suspend	ALLOWSUSP	2
Allow_terminate	ALLOWTERM	2
Allow_text_edit	ALLOWTEXED	2
Allow_view	ALLOWVIEW	1
Answer_length=	ANSLEN=	1
-Automatic_caseid_increment	-AUTOINC	1
Auto_punches	AUTOP	3
Auto_return	AUTORET	2
Backup_here	BACKUPHERE	2
Backup_where_at	BACKUPWHEREAT	2
Blank_lines=	BLANKLINE=	3
Block	BLOCK	5
Card_id_format=	CARD=	1
Case_id=	ID=	1
Case_length=	LEN=	1
Category	CAT	6
Character_set	CHARACTERSET	4
Check_column_overlap	CHKOVERLAP	4
-Check_file	-CHK	1
Column	COL	5
Column_kick	COLKICK	5
Comment	COM	1
Comment	COM	2
Count_as_complete=	COUNTASCOMPLETE=	1

Item	Abbreviation	Reference
Dashboard	DASHBOARD	1
Data_location_required	LOCREQ	1
Data_outside_loop_ok	DATAOUTSIDELOOPOK	1
-DB_file	-DB	1
Default_numeric_minimum	DEFAULTNUMMIN	4
Display	DISP	7
Do_mentor	DOMENTOR	2
Do_variables	DOVAR	2
Echo_cats=	ECHO=	1
Echo_cats	ECHOCAT	2
Edit	EDIT	7
End_after_quit	ENDAFTERQUIT	2
End_block	ENDBLOCK	5
End_loop	ENDLOOP	7
End_remove	ENDREM	4
End_resume	ENDRES	2
End_rotate	ENDROT	2
End_special	ENDSPECIAL	2
End_suspend	ENDSUSP	2
Error_beep	BEEP	1
Error_leadin	ERRORLEADIN	2
Error_msg	ERRORMSG	2
Error_stop	ERRSTOP	4
Expression	EXP	7
Failed_resume=	FAILRESUME=	1
Field	FLD	6
Field_sameas_use_original_width	FIELDSAMEASUSEORIGWID	4
Fix	FIX	2

Item	Abbreviation	Reference
Flag_disallowed_cats	FLAG	1
Fone_text_length=	FONETEXT=	1
Generate	GEN	7
Goto	GOT	7
Group	GRP	2
Hard_code	HARDCODE	4
-Hardcopy file	-HRD	1
Hard_copy	HARDCOPY	4
All	ALL	
Date	DATE	
Language=	LANG=	
Lines=	LINE=	
Page	PG	
Page_length=	PGLEN=	
Page_title=Top/Bottom	PGT=TOP/BOT	
Printer=	PRT=	
Show_cat_as	SCATA	
Show_cat_ns	SCATN	
Show_labels	SLAB	
Show_logic	SLOGIC	
Show_qq_nums	SQQNUM	
Show_samas	SSAMEAS	
-Help	-Help	2
Help_cat	HELPCAT	
Help_display	HELPDISP	
Help_echo_multi	HELPECHOM	
Help_echo_single	HELPECHOS	

Item	Abbreviation	Reference
Help_edit	HELPED!T	
Help_fld	HELPFLD	
Help_fldx	HELPFLDX	
Help_grid	HELPGRID	
Help_hilite_multi	HELPHILITEM	
Help_hilite_single	HELPHILITES	
Help_num	HELPNUM	
Help_reset	HELPRSET	
Help_text	HELPTEXT	
Help_var	HELPVAR	
Help_var_l	HELPVARL	
Help_var_n	HELPVARN	
Help_on_top	HELPONTOP	2
Hide_all	HIDEALL	4
Highlight_cats	HILT	2
Hight_point	HIPT	5
Include	INCLUDE	4
Info_between	INFOBETWEEN	1
Interviewer_logging=	LOG=	1
Keep_response_codes	KEEPRESPCODE	2
Language	LANGUAGE	2
Loop	LOOP	7
Make_names	MAKENAMES	1
Make_spec_files	MSPEC	4
Maximum_consecutive_backups	MAXCONSECBACKUPS	2
Maximum_grid_questions	MAXGRIDQ	1
Maximum_labels=	MAXLAB=	1
Maximum_qfile_size	MAXQFILESIZE	1

Item	Abbreviation	Reference
Maximum_question_size=	MAXQSTSIZE=	1
Maximum_quota_number=	MAXQ=	1
Mentor_cln_file	MENTORCLN	4
Mentor_def_file	MENTORDEF	4
Mentor_tab_file	MENTORTAB	4
Modify_case_id	MODID	1
Modify_fone_file	MODFONE	1
Modify_quota_file	MODQUO	1
Monitor	MONITOR	1
Next_case_id=	NEXTID=	1
Number_of_cases=	NUMCASES=	1
Numbering	NUMBER	4
Numeric	NUM	6
Numeric_width_required	NUMWR	1
One_interview	ONEINT	1
Password=	PASS=	1
Phone	PHO	9
Prepare	PREP	8
Compile	COMPILE	
mentor_specs	MENTOR	
survent_specs	SPEC	
Cleaning_specs	CLN	
Persee_specs	PERSEE	
Quantum_specs	QUANTUM	
-specs	-SPEC	
Spss_specs	SPSS	
Disk_based_response_list	DBR	

Item	Abbreviation	Reference
Make_spec_files	MKFILE	
Sss_xml_specs	SSS	
Qff_file_name=	QFF=	1
-Qsp_file	-QSP	1
-Quota_file	-QUO	1
Quota	QUO	7
Read_names_quotas	READNAMEDQUOTA	2
Redo_special_block_on_resume	REDOSPECIAL	1
Remove	REM	4
Reset	RSET	7
Response_list_columns=	RESPONSELISTCOLUMNS=	2
Restore_column	RESTORECOL	5
Resume	RES	2
Resume_here	RESUMEH	2
Resume_where_at	RESUMEW	2
Rft_cat_01	RFTCAT01	5
Rft_cat_punch	RFTCATP	5
Rft_cat_response	RFTCATR	5
Rft_cat_spread	RFTCATS	5
Rft_on	RFTON	5
Rft_save_loops	RFTSAVEL	5
Rft_unwind_codes	RFTUNWINDC	5
Rft_unwind_loops	RFTUNWINDL	5
Rotate	ROT	2
Rotate_order	ROTORDER	2
Rotate_seed_width	ROTSEEDWID	1
Save_column	SAVECOL	5
Screen_lines=	SCREEN=	1

Item	Abbreviation	Reference
Set_quota	SETQUO	4
Set_interviewer_capabilities=	SETINTVCAP=	4
Show_last_response	SHOWLASTRESPONSE	1
-Show_question_labels	-SHOWQLAB	1
-Show_question_labels	-SHOWQLAB	2
Spec_width=	SPECWID=	1
Special (block)	SPECIAL	2
Special (information)	SPC	7
Spl_data_locations_ok	SPLOK	1
Start_new_card	STARTNEWCARD	4
-Summary_file	-SUM	1
Suspend	SUSP	2
Target	TARG	4
Test_only	TESTONLY	1
Text	TEX	6
Text_help	TEXHELP	1
Text_start=	TEXT=	1
Time_questions	TIMEQUESTIONS	1
Time_on_screen=	TIMEONSCREEN=	1
Time_zone=	TZ=	1
Triple_quotas	TQ	1
TR_file_directory=	TRFDIR=	1
Valid_widths_required	VALIDWIDREQ	1
Variable	VAR	6
View/Viewx	VIEW/VIEWX	4
View_quota	VIEWQUOTA	2
Work_start	WORK=	1

C

APPENDIX C: SOUNDSURVENT SPECS

These sample specs (SND2.QPX) are included as part of the installation of SoundSurvent. This is a basic interviewing questionnaire using SOUND questions. The Coding mode questionnaire, SND3.QPX, also is included as part of the SoundSurvent installation, but it is not printed here.

```
~Comment
This is a basic sound questionnaire in column free mode, to be
used later with a coding mode questionnaire. This file
contains sound subtypes R, P, S,and C.
>Define @Int \I(Press <Return> to Continue)\E
>Define @Text \I(Enter Response and Press <ESC> to Continue)\E
>Define @Record \F(Now Recording)\E
>Define @Stop \I(Press <Return> to Stop Recording)\E
>Define @Play \I(Press <Return> to Begin Playback)\E
>Define @PlayStop \I(Press <Return> at the End of Playback)\E
>Purgesame
~Prepare Compile Specs
[Widgt,400,"Sound Survent Demo -- Widgets",TextStart=3/1]
{!If SoundServer()
!Goto Intro }
{!If NOT(SoundServer()) ‘’try to reestablish communications
!SOUND,X}
{!If SoundServer)_
!GOTO, Intro }
{ SSWarn:
The Sound Server is not currently running. If you were running
this as a sound-exclusive questionnaire, you would want to
terminate at this point (before picking up a phone number, if
running with a phone system).
Otherwise, you can use text questions as an alternate way of
collecting open ends without a Sound Server.

@Int

!Disp }
{ Intro:
Hello, my name is _________ and I'm calling from Widget
```

```
Marketing Research, a very important and prestigious opinion
polling company. Today, I'd like to ask you a few questions
about widgets.

@Int

!Disp }
{ Q1:
1. What brand of widgets do you use most?
!FLD
1 Acme Widgets
2 Adam's Widgets
3 David's Widgets
4 Gidget's Widgets
5 McWidgets
6 Widget Masters
7 Widgets of Steel 2000 }
{!If SoundServer() ''Skip over the standard text
!Goto Q2Disp } ''question if the Sound Server
            ''is working
{ Q2NoSnd:
2. Why do you use \:Q1: the most?

      @Text

!Text,B,1,60 }
{!If Not(SoundServer()) ''Don't ask the Sound question
!Goto End }  ''if the Sound Server isn't working.
''Since there are only questions
''about sound information remaining,
''skip to end of interview.
{ Q2Disp:
2. Why do you use \:Q1: the most?

      @Record

!Disp,2 }
{ Q2Sound:  ''Puts the Sound Server in
!Sound,R }  ''Record mode
{
\A
```

```
      @Stop

!Disp }
{!Sound,S }  ''Stops Recording
{ Playname: .12 Hide
!VAR }
{!Rotate,S }
>Repeat $a="Music","Parrot"
{!Group }
{ [Playname]  ''Picks a sound file to be
$a.RHT       ''played back
!SPC,9 }
{!Goto Out }
>Endrepeat
{!Endrotate }
{ Out:
!Goto }
{ Q3:
3. Next, I would like to play you a short sound recording. Here
it is....
                  Playback Control
                  ======== =======
      7 = Softer                    8 = Slower
      * = Louder                    0 = Faster

      @Play
!Disp }
{!Sound,C,Playname.30 }  ''Playback with limited keypad
                         ''control (7/* pair, 8/0 pair)
{
\A
      @PlayStop
!Disp }
{!Sound,S }  ''Stop playback
{ Q4:
4. Now, can you tell me what you thought about that sound bite?

      @Record

!Disp,2 }
{ Q4Sound:   ''Record mode
!Sound,R }
```

```
{
\A

      @Stop

!Disp }
{!Sound,S }        ''Stop recording
{ PBQ2:
Here's what you said when we asked you about widgets:

      @Play

!Disp }
{!Sound,P,Q2Sound }   ''Playback of file name stored in
                        ''question Q2Sound
{
\A

      @PlayStop

!Disp }
{!Sound,S } ''Stop Playback
{ PBQ4:
      Here's what you said when we asked you about the sound
      bite...

      @Play

!Disp }
{!Sound,P,Q4Sound }  ''Playback of file name stored
                        ''in question Q4File
{
\A
      @PlayStop

!Disp }
{!Sound,S }  ''Stop Playback
{ End:
This is the end of the questionnaire. Thank you for your time.
\I(Press <Return> to End Interview)\E
!Disp }
~End
```

APPENDIX D: BLOW ERROR MESSAGES

D

Some programming errors cannot be captured during a compile of the PREPARE specifications. There are cases when such an error will cause Survent to terminate the interview in progress and quit. If this occurs, an error message will be displayed with a description of the cause of the problem, and the number of the question on which the error occurred. These messages are called blow error messages. Here is an example blow error message:

```
Question too big or positioned off screen at --> Chevrolet <--
FATAL SYSTEM ERROR AT QUESTION #2.50
Please call your supervisor/questionnaire designer for help
and report error #101.
<press <Return> to continue>
```

In addition, Survent will write out (up to the point of the abort) the data case that aborted into a separate directory (DOS, UNIX) named <study>.B_. If the group does not exist, Survent will create it. You can then use this file to try to track down the problem that caused the abort, using VIEW mode of Survent, or CLEANER, to check the responses to that point.

The file LOG<intvid> is saved whenever an interviewer aborts with a blow error. This is to help you in debugging the problem. The LOG file contains a record of all answers typed by the interviewer. Note that this file is cumulative, so the Blow information will appear at the bottom of the file if it already exists. The file will be saved in the CFMCCFG directory (usually \cfmc\ipcfiles in DOS/UNIX).

Blow errors are problems that have been identified by our staff. Other unidentified problems may occur, though. For instance, it is possible to get error messages regarding stack overflow, related to not enough memory on your computer, or due to your computer network or operating system.

These problems will usually require some modification of your computer, but you may want to contact CfMC anyway to see if we have any suggestions that may help eliminate the problem.

Blow errors have one of four causes:

1 Questionnaire designer mistakes: A specification error has been made which caused Survent to abort. These problems must be fixed by the questionnaire designer. These problems are indicated below as "Q'naire" problems. Often they can be found by running Random Data Generation to try to simulate all interviewing possibilities.

2 Survent programming bugs: These are problems we have not yet been able to diagnose yet in the software. You should call us immediately if this occurs: we will need with a good description of what occurred at your site and the files that caused the problem. With this information, we should be able to fix the problem. These are indicated below as "CfMC."

3 FILE CORRUPTION: The questionnaire file, phone file, or related files have been corrupted. This can only be determined by the CfMC staff. These are indicated as "Corrupt xx File" below.

4 SYSTEM: There is some problem with the system (file not available) or the network sending messages or data to/from the SUPERVISOR. These are indicated below as "System".

In the event that Survent blows up, interviewers MUST NOT CLEAR THE SCREEN. They should contact the supervisor and show them the screen. If the supervisor is not available, they should write down the error number, question number, name of the data file saved, and associated messages. Otherwise, the supervisor should record this information.

The supervisor should be able to tell from the message itself or may need to a list of numbers below so that they can determine whether the problem can be fixed by the questionnaire designer, or whether CfMC Customer Service must be called.

SERVER CAN SEND BLOW ERRORS VIA E-MAIL

The Server can send e-mail messages if blows occur.

When the CfMC study server gets a Survent blow, it looks for the file "<studycode>_blow.csh"in the $CFMCCFG directory (usually ${CFMC}ipcfiles) and if that exists, it is executed. If not, it looks for "<studycode> _blow.sh". Otherwise, it looks for ${CFMCCFG}studyblow.sh, and if that exists, it executes that.

If, for some reason, it doesn't know the study code, it uses the study code "unknown" and tries to execute "unknown.sh". The environment variable CFMCBLOW is also created, which contains:

"BLOW at session # from ldev #"

This can be referenced in the shell script.

If there is text string associated with the blow, a second environment variable CFMCBLOWTEXT is created with that text which could be displayed by the script.

The environment variable CFMCBLOWSTUDYCODE is also created, so you can have a generic studyblow.sh file that shows which study code blew up.

Here is an example studyblow.sh 'csh' script:

```
#!/bin/csh

echo "Blow $CFMCBLOW on study $CFMCBLOWSTUDYCODE,

Text: $CFMCBLOWTEXT\n">>xx
```

Error #	Message	Type
01	GET_NEXT_ROTATE_QUESTION: Failed to find question	
02	fs_interview: can't find next question to execute	
03	PUTANSWER: answer array overflow. Increasing the ANSWER_LENGTH value on your header statement is necessary.	Q'naire
04	Generic network error. The station has lost contact with the server.	System
05	Failed to construct clean resume file -- Sorry	

Error #	Message	Type
12	EVALUATE: vcol out of range CfMC	CfMC
13	EVALUATE: string compare width > 70	
14	EVALUATE: comparing different length strings	
16	DO_RESET: reset error (rset to a question never asked)	CfMC
17	GETANSWER: no answer in answer array	CfMC
18	Resume didn’t back up to resume at question number	Q’naire
24	EVALUATE: op function calls unimplemented CfMC	CfMC
25	EVALUATE NUMITEMS gets negative stack	
32	Journal file read error	CfMC
60	Hit ENDROTATE (-513) in setrotpoint	Q’naire
80	Before re-executing interview, could not open new journal file	
81	While re-executing interview, the qfile is BAD	
96	Can’t find endrotate question	Q’naire
97	Rotate with bad rotate header	
99	Unable to get this question	Q’naire
100	Unable to get question in this loop	Q’naire
101	Unable to put question on the screen or specified box	Q’naire
102	Overflow on FLD question with !GEN,O	
103	Blanking text	
104	Disk-based-recode block has too many lines	
105	GOTO question goes to a label that we can't find	
106	CAT,I refers to different width response or not CAT	Q’naire
107	Disk-based-recode table with no matching response	
108	FLD,A CAT,A with no corresponding data The data that was to be recoded isn’t in the table.	Q’naire
109	VAR,A with no corresponding data Don’t have data in a VAR,A with a required minimum length.	Q’naire
110	FLD, disk-based recode, can’t find response Width of field too small for DBR response code	Q’naire

Error #	Message	Type
111	Do_edit_question, target question not VAR or TEXT question	Q'naire
112	Do_edit_question trying for a question and not finding it	Q'naire
113	Trying for a numbered quota larger than [MAX_QUOTA_NUMBER]	Q'naire
114	Too many numbered quotas that are not incremented NOW	Q'naire
*115	Nseval - numbered quota < 1	Q'naire
116	Can't access quota file The file studyname.QUO is not accessible to the user Check the variable CfMCQUO	System
117	Fld GEN A (add) error	
118	Endgrid within rotate	
119	Maximum less than minimum in NUM	
120	Zap_text - text pointer points out of text area	Q'naire
121	Zap_text - back pointer not point to pointer. Can't backup over a GEN,U.	Q'naire
122	Zap_text - next back pointer out of data area	Q'naire
123	Got a SURVSUPR TYPE (QFF name) message in the middle of interviewing	
124	Got an unsolicited phone record. Something is wrong.	
125	Executing special block of questions, failed to get next question	
126	Executing special block of questions, hit a ROTATE or LOOP	
127	Save response, and last journal record does not match question.	
*128	Fillin_sameas_recodetable, can't find question pointed to which	
129	Executing special block, can't find the end block marker	
130	Send_rolm (SPL code 235) in fstartup.C	
131	Send_rolm (SPL code 236) in fstartup.C	
132	Send_rolm (SPL code 237) in fstartup.C	
133	Send_rolm (SPL code 238) in fstartup.C	

Error #	Message	Type
134	Send_rolm (SPL code 239) in fstartup.C	
135	Send_rolm (SPL code 240) in fstartup.C	
136	Send_rolm (SPL code 241) in fstartup.C	
137	Send_rolm on DOS or UNIX	
138	!ENDGRID with uninitialized grid -- did you skip into grid?	
139	Error moving to question in !GRID	
140	More than MAX_GRID_QQS (200+) in grid	Q'naire
141	QREF put greater than max specified	
142	Text pointer put in existing data	Q'naire
143	Data put on exiting text pointer	Q'naire
144	Recode item beyond memory	
145	Unasked question in websurvent grid	Q'naire
146	Unasked question in grid – must use !grid,r	Q'naire
147	!grid,r in websurvent	Q'naire
148	QREF error – Get with no eval	
149	Nextexcol bad set by zap_text	
150	Child_resume with bad response	
151	CFMCRESUME environment not set properly	
152	DBR, error in terminal code or text, or bad data in view	
153	Text length in phone file less than text length in questionnaire	
154	Question width > Maxdatawidth	
155	Evaluate width > Maxdatawidth	
156	RSET in viewmode is fatal	
157	gen,r exceeded available wid	
158	odbc resume failure	
159	QREF error -- no qref array	
160	QREF error -- too many items	
161	QREF error -- too many blocks	
162	fatal core (memory) error	

Error #	Message	Type
201	Getplocation got a bad location (<=0)	
202	Getlocwid offered a label which was not to be found	
203	Trying to do phone stuff in standalone mode	
204	Trying to do a VAR,P with no phone system set up	
205	Picked up case with bad text area. No way to continue	
206	Attempted open data file ‘read only’ failed	
207	SPC,P (read specific case) with a case already in hand	
222	Can’t verify this file Q’naire or Corrupt Data File	Q’naire or Corrupt Data File
223	Data file length is different than QFF file case length	Q’naire
224	Failed to open DOS data file	
225	Failed to open DOS data file	
226	Trying to read or write a local case when we're attached to a Server	
300	Backing up through !CALL: journal history corrupted	
301	Found wrong label trying to back into Q_CALLed trailer	
302	Saved wrong length Survent control structure in Q_CALL	
303	Level difference not 1 between lower Q_CALL	
304	Level difference not 1 between higher Q_CALL	
305	Failed to restore previous questionnaire in trailer Q_CALL	
306	Wrong label for saved answer array in Q_CALL	
307	Wrong label for saved data in Q_CALL	
308	Caselen and ms->prephead.case_len don't match in Q_CALL	
309	Case ID not filled in at Q_CALL	
310	Can’t open Q_CALL data file	
311	Child_level greater than MAX_CHILD_LEVEL (5)	
312	Can't restore DATACASE structure in return_from_call	
313	Level difference not 1 between lower Q_CALL	
314	PHONE,A attempted with no Server	
315	Dialer connection failed	

Error #	Message	Type
316	Error attempting to delete child data record	
317	running in HTML mode, called nextcardc().	
318	Tried to get default phone status screen after it was disabled	
319	Failed to get GRID qq in execute_cgi_response	
320	Giant runaway? Journalfile too big	
321	Highlightcats can only work to a terminal screen	Q'naire
322	Gen,d failure	
323	!call don't return inside child	
400	Endless loop	
401	Child questionnaire blew, so I do too!	
402	failed qwin_ask in s_grid	
403	failed qwin_ask Q_DISP	
404	failed qwin_ask Q_ENDGRID	
405	failed qwin_ask Q_TEXT	
653	qwindow text	
654	text box error	
897	getting case from nonindexed data file	
898	message size > 100000 too large for websurvent	
899	failed to send message to stdysrvr	
900	Network error	System
901	Child questionnaire doing SPC,P or Fonesaysdatarec	
902	Network error - unable to send data case/phone record to CFMC server	System
903	Network error - send a case with different length than master file	System
904	Network error - no phone record to put suspend name into.	
905	PHONE,M to set market, but no markets in file	Q'naire
906	PHONE,G/P/9 with no phone record loaded	Q'naire

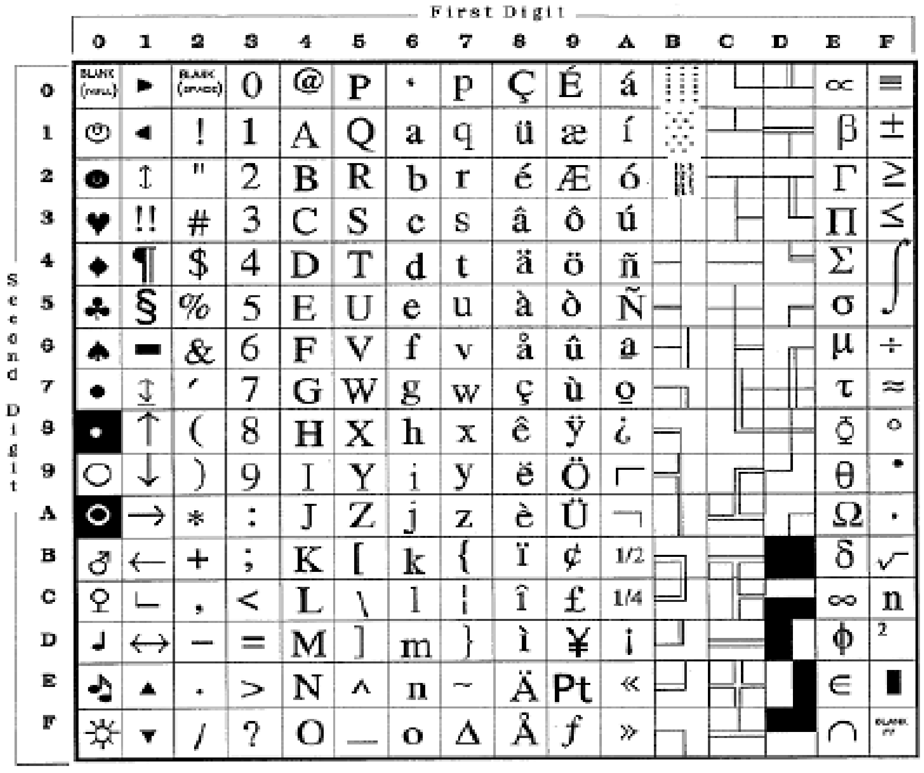
Error #	Message	Type
907	PHONE,D or PHONE,5 with dialer but not preview mode	Q'naire
908	Callback days ahead, shop not open on target day	Q'naire
909	Callback days ahead, no good time on target day	Q'naire
910	Client_abort for some reason	
911	Abnormal exit thru exit_function, possibly no quotafile	
912	websurvent failed to get phone record at startup, see server log	
913		
914		
915	child quotas are different than parent (without –quota_crc_check)	Q'naire
916	validator failed to pick up case	
917	rdgc qq exceeded max allowed	
918	rdgc error	
919	not enough questions matched in rdgc	
920	error parsing !smart_rdg command	
922	server says blow	
923	Fatal error with template file	
926	language not in header	
1106	can't open quotafile	
1107	Quota (name gotten from data) doesn't exist	Q'naire
1108	No allowable response to a CAT/FLD,C/D/I question	Q'naire
9901	callback hour when shop not open	Q'naire
9902	Programmatic timed callback scheduled for impossible date/time	Q'naire

APPENDIX E: SPECIAL CHARACTERS

E

Hexadecimal values may be used to display a variety of special characters and symbols in question text. (See 2.5.2 TEXT ENHANCEMENTS AND GRAPHIC CHARACTERS for information on how to use these graphics.)

GRAPHIC CHARACTERS



First Digit

Second Digit

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	BLANK (NULL)	►	BLANK (SPACE)	0	@	P	`	p	Ç	É	á	░	└	╨	α	≡
1	☺	◄	!	1	A	Q	a	q	ü	æ	í	▒	┴	╤	β	±
2	☻	↕	"	2	B	R	b	r	é	Æ	ó	▓	┬	╥	Γ	≥
3	♥	‼	#	3	C	S	c	s	â	ô	ú	│	├	╙	Π	≤
4	♦	¶	$	4	D	T	d	t	ä	ö	ñ	┤	─	╘	Σ	⌠
5	♣	§	%	5	E	U	e	u	à	ò	Ñ	╡	┼	╒	σ	⌡
6	♠	▬	&	6	F	V	f	v	å	û	ª	╢	╞	╓	µ	÷
7	•	↨	'	7	G	W	g	w	ç	ù	º	╖	╟	╫	τ	≈
8	◘	↑	(	8	H	X	h	x	ê	ÿ	¿	╕	╚	╪	Φ	°
9	○	↓	)	9	I	Y	i	y	ë	Ö	⌐	╣	╔	┘	Θ	∙
A	◙	→	*	:	J	Z	j	z	è	Ü	¬	║	╩	┌	Ω	·
B	♂	←	+	;	K	[	k	{	ï	¢	1/2	╗	╦	█	δ	√
C	♀	∟	,	<	L	\	l	¦	î	£	1/4	╝	╠	▄	∞	n
D	♪	↔	-	=	M	]	m	}	ì	¥	¡	╜	═	▌	φ	2
E	♫	▲	.	>	N	^	n	~	Ä	Pt	«	╛	╬	▐	∈	■
F	☼	▼	/	?	O	_	o	Δ	Å	ƒ	»	┐	╧	▀	∩	BLANK FF

Enter a \^ and the two-digit hexadecimal value coresponding to the desired symbol. For example;

```
THAT CONCLUDES OUR INTERVIEW. THANK YOU. \^01
```

In this example, the text will be followed by a smiling face (hexadecimal value 01).

NOTE: Some characters only display on a certain operating systems. See the example file SCREN.QPX in CFMC\SURVENT (or CSURVENT.CFMC) for a setup to test this.

Certain characters have special meanings in CfMC software. Some characters have the same type of meaning in different usages. When a character can have more than one meaning, it is also noted.

ADDITIONAL NOTE: CfMC recommends that you use your browser to search for information regarding Graphic Characters for your particular Operating System.

Prepare

Name	Character	Syntax	Description
Tilde	~	~PREPARE	Lead-in character for system level commands, specifically ~PREPARE options and ~END specifications. Must start in column 1 and immediately precede keyword.

Study Header

Name	Character	Syntax	Description
Brackets	[ ]	[studyname]	Enclose the study header statement.
Comma	,	CASE_ID=1.5,-CHECK	Used to separate options, or as placeholder for non-specified case length or comment options.
Double quote	" "	"study #123"	Delimiter for study comment.
Ampersand	&	ERROR_BEEP,&	Used as a continuation character — to continue the study header options on the next line.

Compiler Statements

Name	Character	Syntax	Description
Braces	{ }	{!ROTATE}	Begin/end of a compiler statement.
Exclamation point	!	{!ROTATE}	Starts a compiler statement keyword.
	!-	{!-AUTO_PUNCHES}	Turn off a compiler option.
Comma	,	{!ROTATE,S,1}	Separates parameters

Question Statements

Name	Character	Syntax	Description
Braces	{ }	{!First	Begin/end of a question statement.
Colon	:	{Test:	Ends a question label
Crosshatch	#	#2.50	Precedes a question number on a question label line.
Brackets	[ ]	[1/23]	Enclose a data location on a question label line
Ampersand	&	!IF Last(1) &	Used as a continuation character to continue !IF statements to the next line.
Backslash	\	\Bbold\E	Used as a start character for text control characters.
Exclamation point	!	!CAT	Starts a question statement type or condition (e.g., !IF).
Comma	,	!CAT,A,5	Used as a delimiter between options in an option list. Used on question type and response list control options.

Response List

Name	Character	Syntax	Description
Parenthesis	( )	(SKIPTO next)	Used around the response list control options.
Dash	-	(-Y)	Used to mark response exclusivity on multiple response questions. These items will not be rotated if question subtype R is specified.
Caret	^	^^text	Used as a placeholder for <blank> as a response code or to indent response text.
Brackets	[ ]	[SAMEAS last]	Start/stop SAMEAS response list reference within a question statement.
Equal sign	=	= CARTYPES:	Used to format the response list, equal sign responses cannot be chosen by the interviewer, and their position in a rotated list remains fixed.
Exclamation point	!	(!)	Fixes a response item in place when sorting or rotating.

PREPARE condition/expression statement special characters

These are used within the IF condition on questions, and on EXP statements:

Type	Symbol(s)	Syntax	Description
Math characters	+ - / *	QQ3 + 1	Add, subtract, divide, and multiply.
Math conditionals	> = <	QQ3 >= 1	Combine for greater than, equal to, and less than numeric references.

Type	Symbol(s)	Syntax	Description
Parenthesis	( )	(5+2)/3	Separate logical pieces of conditions or expressions.
		NUMITEMS (QQ3)	Surround argument for function statements.
		QQ3(1)	Surround response code list references from CAT and FLD questions.
Brackets	[ ]	[2/34]	Surround data location references.
Dollar sign	$	Name$="abc"	Specifies a string variable reference.
Double quote	“ “	Name$="abc"	Delimiter for strings.
Period	.	QQ3(1.5)	Specifies a range of responses from a response list.
		QQ3(1,2,5)	Delimiter for list of responses from a response list.
Comma	,	MAX(QN10,QN11)	Separates items in functions.
Dash	-	QQ3(1-5)	Specifies a range of responses from a response list.
Not equal	< >	QQ17(< >03)	Answer *cannot* be code(s) listed.

Data Location References

Name	Character	Syntax	Description
Brackets	[ ]	[2/45]	Surrounds data location references.
Slash	/	[2/45]	Separates a record number reference from the column number (within the record).
Period	.	[2/45.5]	Separates a location specification from the length.
		[2/45^1.5]	Specifies binary punch type data reference.
Caret	^	[2/45^1]	Specifies binary punch type data reference.
Crosshatch	#	[2/45#1]	Specifies numeric category data reference.
Dash	-	[1/10.3#1-999]	Specifies a range in a number reference.
Dollar Sign	$	[2/45.5$]	Specifies string type data reference.
Comma	,	[2/45^1,5]	Specifies separate responses in a binary punch list.

System

Name	Character	Syntax	Description
Single quotes	‘ ‘	‘ ‘Note here	Two single quotes denote a comment that is not interpreted by the program. Can be at the beginning of the line or after commands on a line.
Greater-than	>	>PURGE_SAME	Lead-in character for meta commands. Can be used anywhere between other statements; must be at beginning of line.
At sign	@	@A	Defines character for keyword substitution.

File Name References

Name	Character	Syntax	Description
Backslash	\	\Cfmc\Demo	Delimiter between directory references in DOS.
Frontslash	/	/CFMC/DEMO	Delimiter between directory references in UNIX.
Period	.	Bank.qpx	Delimiter between file name and extension (DOS, UNIX).
Exclamation point	!	!Cfmc!Demo	Used around system variable references in the file name.
Ampersand	&	[1/10.3#1-999]	When referencing files within a program, used to access a file, and describe the type of file (&=ASCII file, &&=dbentry, &&&=questionnaire file).
		&-	As above, but don't display contents of file to list file.
Quotes	" "		Use around names with more than 8 characters or with special characters (UNIX, DOS)
Dollar sign	$		Says to use this file name exactly. Don't add CfMC standard extensions.
Asterisk	*		Wild card in DOS and UNIX
At sign	@		Wild card; will match with any number of any characters.
Crosshatch	#		Wild card; will match with one alphabetic character.
Question mark	?		Wild card; will match with one alphabetic character.

APPENDIX F: LANGUAGE CONVERSION

When you compile your specs, you have the opportunity to get various other files created. *2.3.1 COMPOSING AND COMPILING QUESTIONS IN PREPARE* talks about this process. What is listed below is sample output from the RRUNR questionnaire (see *Appendix A: SAMPLE SPECIFICATIONS*) when it is compiled using the different keywords.

NOTES:

- When doing an SPSS compile, you will get a warning message for any multiple-response question, as well as any CAT question using more than one column.

(WARN #5321) SPSS variable from CAT with more than one item possible

(WARN #5320) SPSS variable from CAT of more than one column ... check its values

- You can create complete sets of the files shown below by compiling \CFMC\SURVENT\RRUNR.QPX and using the appropriate keyword on the ~PREPARE COMPILE line. Shown below are representative selections from the files.

File type	Page
Uncle TAB file	
Triple-S XML	

Mentor CLN file:

```
QN1:
edit QN1
QN2A:
edit QN2A
QN2B:
edit QN2B
QN5G:
edit QN5G
QN6:
edit QN6
QN6A:
if (QN6(1))
edit QN6A
endif
QN6B:
if (QN6(1) and not (QN6A(1)))
edit QN6B
endif
QN8:
edit QN8
QN15:
edit QN15
QN20:
edit QN20
QN21:
edit QN21
DEMO1:
edit DEMO1
LAST:
edit LAST
```

Mentor DEF file:

```
tabset= { qn1_z:
title=:
   Q1. How much do you agree with the following
```

```
    statement:
    The fast food at Road Runners is worth what I pay for
    it.}
stub=:
     (5) Completely agree
     (4) Somewhat agree
     (3) Neither agree nor disagree
     (2) Somewhat disagree
     (1) Completely disagree
    Don't Know/Refused to answer
     [stat] Mean
     [stat] Standard deviation
     [stat] Standard error}
''qn1(5/4/3/2/1/0)
row=: [1/6.1^5/4/3/2/1/10] &
     $[mean,std,se] [1/6.1 *ranges=1-5]
}
tabset= { qn2a_z:
title=:
     Q2a. Please rate the following characteristics:
     The quality of the food.}
stub=:
    (5) Very good
    (4) Good
    (3) Neither poor nor good
    (2) Poor
    (1) Very poor
    Don't know/refused to answer
    [stat] Mean
    [stat] Standard deviation
    [stat] Standard error}
''qn2a(5/4/3/2/1/0)
row=: [1/7.1^5/4/3/2/1/10] &
$[mean,std,se] [1/7.1 *ranges=1-5]
    }
tabset= { qn2b_z:
title=:
   Q2B. Please rate the following characteristics:
   The quality of service.}
'' SAMEAS QN2A
stub=qn2a_s
''qn2b(5/4/3/2/1/0)
```

```
row=: [1/8.1^5/4/3/2/1/10] &
   $[mean,std,se] [1/8.1 *ranges=1-5]
}
tabset= { qn4_z:
title=:
   Q4. About how much do you pay per visit for
   Road Runner's fast food - that is,
   not including entertainment?}
stub=:
   5-13
   14-20
   21-28
   29-50
   RF
   [stat] Mean
   [stat] Standard deviation
   [stat] Standard error}
row=: [1/15.2 # 5-13/14-20/21-28/29-50/"RF" ] &
$[mean,std,se] [1/15.2 *ranges=5,50,,RF,,]
   }
tabset= { qn5g_z:
title=:
   Q5g. In response to the statement, "Road Runners fast
   food provides good food at realistic and affordable
   rates." Do you . . .}
'' SAMEAS QN1
stub=qn1_s
''qn5g(5/4/3/2/1/0)
row=: [1/26.1^5/4/3/2/1/10] &
$[mean,std,se] [1/26.1 *ranges=1-5]
   }
tabset= { qn6_z:
title=:
   Q6. Has anyone in your household participated in the
   entertainment available at Road Runners during the
   past three months?}
stub=:
   Yes
   No
   Don't Know}
''qn6(1/2/0)
row=: [1/30.1^1/2/10]
```

```
   }
tabset= { qn6a_z:
title=:
   Q6a. Which entertainment was participated in during
   the past three months?}
stub=:
Video games
   Billiards
   Fun House
   Musical Revue
   Dunk the Moose
   Other
   Don't know/refused}
''qn6a(1/2/3/4/5/6/7)
row=: [1/31.1^1/2/3/4/5/6/( exclusive)7]
   }
tabset= { qn6b_z:
title=:
   Q6b. Have you used the video games at Road Runners
   within the past three months?}
'' SAMEAS QN6
stub=qn6_s
''qn6b(1/2/0)
row=: [1/32.1^1/2/10]
   }
tabset= { qn8_z:
title=:
   Q8. From your own experience and knowledge, what do
   you especially like about Road Runners?}
stub=:
   Good service/prompt service
   Dependable/continuous service
   The courteous employees they have/helpful
   I like the food/good food
   Food selection
   Good prices
   Computerized/accurate billing
   Helpful in explaining billing questions
   Good entertainment
   Variety of entertainment
   Nice family atmosphere
   Established place/has been around for awhile
```

```
    Accessible/Available/They're everywhere
    Like everything/good place
    Other
    No problem/No complaints
    Don't know/No answer
    Nothing}
''qn8(01/02/03/04/05/06/07/08/09/10/11/12/13/14/15/22/23/24)
row=: [1/40.2^1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/(exclusive)&
22/(exclusive) 23/(exclusive)24]
   }
tabset= { qn15_z:
title=:
   Q15. What is the highest level of formal education
   completed by the principal wage earner in your home?}
stub=:
   Less than grade school
   Grade school
   Some high school
   High school diploma
   Technical school
   Some college
   College graduate
   Graduate school
   Refused}
''qn15(1/2/3/4/5/6/7/8/9)
row=: [1/50.1^1/2/3/4/5/6/7/8/9]
   }
tabset= { qn20_z:
title=:
   Q20. How many children ages 18 and under are in living
   at  home?}
stub=:
   0-3
   4-5
   6-10
   RF
   [stat] Mean
   [stat] Standard deviation
   [stat] Standard error}
row=: [1/55.2 # 0-3/4-5/6-10/"RF" ] &
$[mean,std,se] [1/55.2 *ranges=0,10,,RF,,]
   }
```

```
tabset= { qn21_z:
title=:
   Q21. Respondent Sex}
stub=:
   Male
   Female
   Not determined}
''qn21(1/2/3)
row=: [1/57.1^1/2/3]
   }
tabset= { last_z:
title=:
   Respondent state code}
stub=:
   Alaska
   Alabama
   Arkansas
   Arizona
   California
   Colorado
   Connecticut
   District of Columbia
   Delaware
   Florida
   Georgia
   Hawaii
   Wyoming}
''last(AK/AL/AR/AZ/CA/CO/CT/DC/DE/FL/GA/HI/&
''WY)
row=: [1/58.2 # AK/AL/AR/AZ/CA/CO/CT/DC/DE/FL/GA/HI/&
   WY]
   }
```

Mentor TAB file:

```
tabset=qn1_z
tabset=qn2a_z
tabset=qn2b_z
tabset=qn5g_z
tabset=qn6_z
tabset=qn6a_z
tabset=qn6b_z
```

```
tabset=qn8_z
tabset=qn15_z
tabset=qn20_z
tabset=qn21_z
tabset=last_z
```

COSI DEF file:

```
Q000.05 CAT 0006 001 01 006 Q1. How much do you agree
with the following statement: ....
R5     00 5 (5) Completely agree
R4     00 4 (4) Somewhat agree
R3     00 3 (3) Neither agree nor disagree
R2     00 2 (2) Somewhat disagree
R1     00 1 (1) Completely disagree
R0     00 0 Don't Know/Refused to answer
Q000.07 CAT 0007 001 01 006 Q2a. Please rate the
following characteristics: The qual....
R5     00 5 (5) Very good
R4     00 4 (4) Good
R3     00 3 (3) Neither poor nor good
R2     00 2 (2) Poor
R1     00 1 (1) Very poor
R0     00 0 Don't know/refused to answer
Q000.09 CAT 0008 001 01 006 Q2B. Please rate the
following characteristics: The qual....
R5     00 5 (5) Very good
R4     00 4 (4) Good
R3     00 3 (3) Neither poor nor good
R2     00 2 (2) Poor
R1     00 1 (1) Very poor
R0     00 0 Don't know/refused to answer
Q000.37 CAT 0026 001 01 006 Q5g. In response to the
statement, "Road Runners fast fo....
R5     00 5 (5) Completely agree
R4     00 4 (4) Somewhat agree
R3     00 3 (3) Neither agree nor disagree
R2     00 2 (2) Somewhat disagree
R1     00 1 (1) Completely disagree
R0     00 0 Don't Know/Refused to answer
Q000.39 CAT 0030 001 01 003 Q6. Has anyone in your
household participated in the ent....
```

```
R1     00 1 Yes
R2     00 2 No
R0     00 0 Don't Know
Q000.41 CAT 0031 001 06 007 Q6a. Which entertainment was
participated in during the ....
R1     00 1 Video games
R2     00 2 Billiards
R3     00 3 Fun House
R4     00 4 Musical Revue
R5     00 5 Dunk the Moose
R6     00 6 Other
R7     00 7 Don't know/refused
Q000.43 CAT 0032 001 01 003 Q6b. Have you used the video
games at Road Runners ....
R1     00 1 Yes
R2     00 2 No
R0     00 0 Don't Know
Q000.59 CAT 0040 002 15 018 Q8. From your own experience
and knowledge, what do you ....
R01    00 1 Good service/prompt service
R02    00 2 Dependable/continuous service
R03    00 3 The courteous employees they hav....
R04    00 4 I like the food/good food
R05    00 5 Food selection
R06    00 6 Good prices
R07    00 7 Computerized/accurate billing
R08    00 8 Helpful in explaining billing qu....
R09    00 9 Good entertainment
R10    00 0 Variety of entertainment
R11    00 X Nice family atmosphere
R12    00 Y Established place/has been aroun....
R13    01 1 Accessible/Available/They're eve....
R14    01 2 Like everything/good place
R15    01 3 Other
R22    01 0 No problem/No complaints
R23    01 X Don't know/No answer
R24    01 Y Nothing
Q000.75 CAT 0050 001 01 009 Q15. What is the highest
level of formal education compl....
R1     00 1 Less than grade school
R2     00 2 Grade school
R3     00 3 Some high school
```

```
R4     00 4 High school diploma
R5     00 5 Technical school
R6     00 6 Some college
R7     00 7 College graduate
R8     00 8 Graduate school
R9     00 9 Refused
Q000.85 NUM 0055 002 0 Q20. How many children ages 18 and
under are in living a....
Q000.87 CAT 0057 001 01 003 Q21. Respondent Sex
R1     00 1 Male
R2     00 2 Female
R3     00 3 Not determined
Q000.89 VAR 0241 035 May I have your name so that my
supervisor can verify th....
Q000.91 TEXT 0311 001 And are there any comments you wish
to make about this q....
Q000.93 FLD 0058 002 01 051 Respondent state code
RAK          Alaska
RAL          Alabama
RAR          Arkansas
RAZ          Arizona
RCA          California
RCO          Colorado
RCT          Connecticut
RDC          District of Columbia
RDE          Delaware
RFL          Florida
RGA          Georgia
RHI          Hawaii
RWY          Wyoming
```

Quantum DEF file:

```
l qn1
ttrQ1. How much do you agree with the following
statement: ....
ttcqn1
*include base
n01(5) Completely agree;                   c=c106'5'
n01(4) Somewhat agree;                     c=c106'4'
n01(3) Neither agree nor disagree;         c=c106'3'
n01(2) Somewhat disagree;                  c=c106'2'
```

```
n01(1) Completely disagree;                 c=c106'1'
n01Don't Know/Refused to answer;            c=c106'0'
n01Not asked;                               c=-
l qn2a
ttrQ2a. Please rate the following characteristics: The
qual....
ttcqn2a
*include base
n01(5) Very good;                           c=c107'5'
n01(4) Good;                                c=c107'4'
n01(3) Neither poor nor good;               c=c107'3'
n01(2) Poor;                                c=c107'2'
n01(1) Very poor;                           c=c107'1'
n01Don't know/refused to answer;            c=c107'0'
n01Not asked;                               c=-
l qn2b
ttrQ2B. Please rate the following characteristics: The
qual....
ttcqn2b
*include base
n01(5) Very good;                           c=c108'5'
n01(4) Good;                                c=c108'4'
n01(3) Neither poor nor good;               c=c108'3'
n01(2) Poor;                                c=c108'2'
n01(1) Very poor;                           c=c108'1'
n01Don't know/refused to answer;            c=c108'0'
n01Not asked;                               c=-
l qn5g
ttrQ5g. In response to the statement, "Road Runners fast
fo....
ttcqn5g
*include base
n01(5) Completely agree;                    c=c126'5'
n01(4) Somewhat agree;                      c=c126'4'
n01(3) Neither agree nor disagree;          c=c126'3'
n01(2) Somewhat disagree;                   c=c126'2'
n01(1) Completely disagree;                 c=c126'1'
n01Don't Know/Refused to answer;            c=c126'0'
n01Not asked;                               c=-
l qn6
ttrQ6. Has anyone in your household participated in the
ent....
```

```
ttcqn6
*include base
n01Yes;                                   c=c130'1'
n01No;                                    c=c130'2'
n01Don't Know;                            c=c130'0'
n01Not asked;                             c=-
l qn6a
ttrQ6a. Which entertainment was participated in during
the ....
ttcqn6a
*include base
n01Video games;                           c=c131'1'
n01Billiards;                             c=c131'2'
n01Fun House;                             c=c131'3'
n01Musical Revue;                         c=c131'4'
n01Dunk the Moose;                        c=c131'5'
n01Other;                                 c=c131'6'
n01Don't know/refused;                    c=c131'7'
n01Not asked;                             c=-
l qn6b
ttrQ6b. Have you used the video games at Road Runners
....
ttcqn6b
*include base
n01Yes;                                   c=c132'1'
n01No;                                    c=c132'2'
n01Don't Know;                            c=c132'0'
n01Not asked;                             c=-
l qn8
ttrQ8. From your own experience and knowledge, what do
you ....
ttcqn8
*include base
n01Good service/prompt service;           c=c140'1'
n01Dependable/continuous service;         c=c140'2'
n01The courteous employees they hav....;  c=c140'3'
n01I like the food/good food;             c=c140'4'
n01Food selection;                        c=c140'5'
n01Good prices;                           c=c140'6'
n01Computerized/accurate billing;         c=c140'7'
n01Helpful in explaining billing qu....;  c=c140'8'
n01Good entertainment;                    c=c140'9'
```

```
n01Variety of entertainment;                  c=c140'0'
n01Nice family atmosphere;                    c=c140'-'
n01Established place/has been aroun....;      c=c140'&'
n01Accessible/Available/They're eve....;      c=c141'1'
n01Like everything/good place;                c=c141'2'
n01Other;                                     c=c141'3'
n01No problem/No complaints;                  c=c141'0'
n01Don't know/No answer;                      c=c141'-'
n01Nothing;                                   c=c141'&'
n01Not asked;                                 c=-
l qn15
ttrQ15. What is the highest level of formal education
compl....
ttcqn15
*include base
n01Less than grade school;                    c=c150'1'
n01Grade school;                              c=c150'2'
n01Some high school;                          c=c150'3'
n01High school diploma;                       c=c150'4'
n01Technical school;                          c=c150'5'
n01Some college;                              c=c150'6'
n01College graduate;                          c=c150'7'
n01Graduate school;                           c=c150'8'
n01Refused;                                   c=c150'9'
n01Not asked;                                 c=-
l qn20
ttrQ20. How many children ages 18 and under are in living
a....
ttcqn20
*include base
val c(155,156);I;0-2;3-5;6-10;Others=rej
l qn21
ttrQ21. Respondent Sex
ttcqn21
*include base
n01Male;                                      c=c157'1'
n01Female;                                    c=c157'2'
n01Not determined;                            c=c157'3'
n01Not asked;                                 c=-
l last
ttrRespondent state code
ttclast
```

```
*include base
n01Alaska;                          c=c(158,159)=$AK$
n01Alabama;                         c=c(158,159)=$AL$
n01Arkansas;                        c=c(158,159)=$AR$
n01Arizona;                         c=c(158,159)=$AZ$
n01California;                      c=c(158,159)=$CA$
n01Colorado;                        c=c(158,159)=$CO$
n01Connecticut;                     c=c(158,159)=$CT$
n01District of Columbia;            c=c(158,159)=$DC$
n01Delaware;                        c=c(158,159)=$DE$
n01Florida;                         c=c(158,159)=$FL$
n01Georgia;                         c=c(158,159)=$GA$
n01Hawaii;                          c=c(158,159)=$HI$
n01Wyoming;                         c=c(158,159)=$WY$
n01Not asked;                       c=-
```

Quantum TAB file:

```
tab qn1                 &ban
tab qn2a                &ban
tab qn2b                &ban
tab qn5g                &ban
tab qn6                 &ban
tab qn6a                &ban
tab qn6b                &ban
tab qn8                 &ban
tab qn15                &ban
tab qn20                &ban
tab qn21                &ban
tab last                &ban
```

SPSS DEF file:

```
DATA LIST FILE='RRUNR.RFT'
/ qn1 6
   qn2a 7
   qn2b 8
   qn5g 26
   qn6 30
   qn6a 31
   qn6b 32
   qn801 40 (A)
   qn802 41 (A)
   qn15 50
   qn20 55-56
   qn21 57
   last 58-59 (A)
.
VALUE LABELS
qn1
5 "(5) Completely agree"
4 "(4) Somewhat agree"
3 "(3) Neither agree nor disagree"
2 "(2) Somewhat disagree"
1 "(1) Completely disagree"
0 "Don't Know/Refused to answer"/
qn2a
5 "(5) Very good"
4 "(4) Good"
3 "(3) Neither poor nor good"
2 "(2) Poor"
1 "(1) Very poor"
0 "Don't know/refused to answer"/
qn2b
5 "(5) Very good"
4 "(4) Good"
3 "(3) Neither poor nor good"
2 "(2) Poor"
1 "(1) Very poor"
0 "Don't know/refused to answer"/
qn5g
5 "(5) Completely agree"
4 "(4) Somewhat agree"
3 "(3) Neither agree nor disagree"
2 "(2) Somewhat disagree"
1 "(1) Completely disagree"
0 "Don't Know/Refused to answer"/
qn6
1 "Yes"
2 "No"
0 "Don't Know"/
qn6a
1 "Video games"
2 "Billiards"
3 "Fun House"
```

```
4 "Musical Revue"
5 "Dunk the Moose"
6 "Other"
7 "Don't know/refused"/
qn6b
1 "Yes"
2 "No"
0 "Don't Know"/
qn801
"1" "Good service/prompt service"
"2" "Dependable/continuous service"
"3" "The courteous employees they have/helpful"
"4" "I like the food/good food"
"5" "Food selection"
"6" "Good prices"
"7" "Computerized/accurate billing"
"8" "Helpful in explaining billing questions"
"9" "Good entertainment"
"0" "Variety of entertainment"
"-" "Nice family atmosphere"
"&" "Established place/has been around for awhile"/
qn802
"1" "Accessible/Available/They're everywhere"
"2" "Like everything/good place"
"3" "Other"
"0" "No problem/No complaints"
"-" "Don't know/No answer"
"&" "Nothing"/
qn15
1 "Less than grade school"
2 "Grade school"
3 "Some high school"
4 "High school diploma"
5 "Technical school"
6 "Some college"
7 "College graduate"
8 "Graduate school"
9 "Refused"/
qn21
1 "Male"
2 "Female"
3 "Not determined"/
```

```
last
"AK" "Alaska"
"AL" "Alabama"
"AR" "Arkansas"
"AZ" "Arizona"
"CA" "California"
"CO" "Colorado"
"CT" "Connecticut"
"DC" "District of Columbia"
"DE" "Delaware"
"FL" "Florida"
"GA" "Georgia"
"HI" "Hawaii"
"WY" "Wyoming"/
.

VARIABLE LABELS
qn1 "Q1. How much do you agree with the following statement: ...."
qn2a "Q2a. Please rate the following characteristics: The qual...."
qn2b "Q2B. Please rate the following characteristics: The qual...."
qn5g "Q5g. In response to the statement, 'Road Runners fast fo...."
qn6 "Q6. Has anyone in your household participated in the ent...."
qn6a "Q6a. Which entertainment was participated in during the ...."
qn6b "Q6b. Have you used the video games at Road Runners ...."
qn801 "Q8. From your own experience and knowledge, what do you ...."
qn802 "Q8. From your own experience and knowledge, what do you ...."
qn15 "Q15. What is the highest level of formal education compl...."
qn20 "Q20. How many children ages 18 and under are in living a...."
qn21 "Q21. Respondent Sex"
last "Respondent state code"
.
```

SPSS TAB file:

```
qn1 "Q1. How much do you agree with the following statement: ...."
qn2a "Q2a. Please rate the following characteristics: The qual...."
qn2b "Q2B. Please rate the following characteristics: The qual...."
qn5g "Q5g. In response to the statement, 'Road Runners fast fo...."
qn6 "Q6. Has anyone in your household participated in the ent...."
qn6a "Q6a. Which entertainment was participated in during the ...."
qn6b "Q6b. Have you used the video games at Road Runners ...."
qn801 "Q8. From your own experience and knowledge, what do you ...."
qn802 "Q8. From your own experience and knowledge, what do you ...."
qn15 "Q15. What is the highest level of formal education compl...."
qn20 "Q20. How many children ages 18 and under are in living a...."
qn21 "Q21. Respondent Sex"
last "Respondent state code"
```

SPSS LAB file:

```
qn1
5 "(5) Completely agree"
4 "(4) Somewhat agree"
3 "(3) Neither agree nor disagree"
2 "(2) Somewhat disagree"
1 "(1) Completely disagree"
0 "Don't Know/Refused to answer"/
qn2a
5 "(5) Very good"
4 "(4) Good"
3 "(3) Neither poor nor good"
2 "(2) Poor"
1 "(1) Very poor"
0 "Don't know/refused to answer"/
qn2b
5 "(5) Very good"
4 "(4) Good"
3 "(3) Neither poor nor good"
2 "(2) Poor"
1 "(1) Very poor"
0 "Don't know/refused to answer"/
qn5g
5 "(5) Completely agree"
4 "(4) Somewhat agree"
3 "(3) Neither agree nor disagree"
2 "(2) Somewhat disagree"
1 "(1) Completely disagree"
0 "Don't Know/Refused to answer"/
qn6
1 "Yes"
2 "No"
0 "Don't Know"/
qn6a
1 "Video games"
2 "Billiards"
3 "Fun House"
4 "Musical Revue"
5 "Dunk the Moose"
6 "Other"
7 "Don't know/refused"/
```

```
qn6b
1 "Yes"
2 "No"
0 "Don't Know"/
qn801
"1" "Good service/prompt service"
"2" "Dependable/continuous service"
"3" "The courteous employees they have/helpful"
"4" "I like the food/good food"
"5" "Food selection"
"6" "Good prices"
"7" "Computerized/accurate billing"
"8" "Helpful in explaining billing questions"
"9" "Good entertainment"
"0" "Variety of entertainment"
"-" "Nice family atmosphere"
"&" "Established place/has been around for awhile"/
qn802
"1" "Accessible/Available/They're everywhere"
"2" "Like everything/good place"
"3" "Other"
"0" "No problem/No complaints"
"-" "Don't know/No answer"
"&" "Nothing"/
qn15
1 "Less than grade school"
2 "Grade school"
3 "Some high school"
4 "High school diploma"
5 "Technical school"
6 "Some college"
7 "College graduate"
8 "Graduate school"
9 "Refused"/
qn21
1 "Male"
2 "Female"
3 "Not determined"/
last
"AK" "Alaska"
"AL" "Alabama"
"AR" "Arkansas"
```

```
"AZ" "Arizona"
"CA" "California"
"CO" "Colorado"
"CT" "Connecticut"
"DC" "District of Columbia"
"DE" "Delaware"
"FL" "Florida"
"GA" "Georgia"
"HI" "Hawaii"
"WY" "Wyoming"/
```

SPSS LPR file:

```
MISSING VALUES qn4 (,RF)
MISSING VALUES qn20 (,RF)
```

UNCLE TAB file:

```
T1 &RJTABLE &TN /
T2 / /&CP QN1 - Q1. How much do you agree with the
following statement: The fast food at Road Runners is
worth what I pay for it.
T3 TABLE &TN      QN1 - Q1. How much do you agree with
the following statement: The fast food at Road Runners
is ...
R TOTAL RESPONDENTS; ALL
R (5) Completely agree;6-5
R (4) Somewhat agree;6-4
R (3) Neither agree nor disagree;6-3
R (2) Somewhat disagree;6-2
R (1) Completely disagree;6-1
R Don't Know/Refused to answer;6-0
R NOT REPORTED;NOT
T1 &RJTABLE &TN /
T2 / /&CP         QN2A - Q2a. Please rate the following
characteristics: The quality of the food.
T3 TABLE &TN      QN2A - Q2a. Please rate the following
characteristics: The quality of the food.
R TOTAL RESPONDENTS; ALL
R (5) Very good;7-5
R (4) Good;7-4
R (3) Neither poor nor good;7-3
R (2) Poor;7-2
```

```
R (1) Very poor;7-1
R Don't know/refused to answer;7-0
R NOT REPORTED;NOT
T1 &RJTABLE &TN /
T2 / /&CP QN2B - Q2B. Please rate the following
characteristics: The quality of service.
T3 TABLE &TN      QN2B - Q2B. Please rate the following
characteristics: The quality of service.
R TOTAL RESPONDENTS; ALL
R (5) Very good;8-5
R (4) Good;8-4
R (3) Neither poor nor good;8-3
R (2) Poor;8-2
R (1) Very poor;8-1
R Don't know/refused to answer;8-0
R NOT REPORTED;NOT
T1 &RJTABLE &TN /
T2 / /&CP         QN6 - Q6. Has anyone in your household
participated in the entertainment
available at Road Runners during the past three m
T3 TABLE &TN      QN6 - Q6. Has anyone in your household
participated in the entertainment available at Road
Runner
R TOTAL RESPONDENTS; ALL
R Yes;30-1
R No;30-2
R Don't Know;30-0
R NOT REPORTED;NOT
T1 &RJTABLE &TN /
T2 / /&CP QN6A - Q6a. Which entertainment was
participated in during the past three months?
T3 TABLE &TN      QN6A - Q6a. Which entertainment was
participated in during the past three months?
R TOTAL RESPONDENTS; ALL
R Video games;31-1
R Billiards;31-2
R Fun House;31-3
R Musical Revue;31-4
R Dunk the Moose;31-5
R Other;31-6
R Don't know/refused;31-7
R NOT REPORTED;NOT
```

```
T1 &RJTABLE &TN /
T2 / /&CP        QN15 - Q15. What is the highest level
of formal education completed by the principal wage
earner in your home?
T3 TABLE &TN     QN15 - Q15. What is the highest level
of formal education completed by the principal wage
earner
R TOTAL RESPONDENTS; ALL
R Less than grade school;50-1
R Grade school;50-2
R Some high school;50-3
R High school diploma;50-4
R Technical school;50-5
R Some college;50-6
R College graduate;50-7
R Graduate school;50-8
R Refused;50-9
R NOT REPORTED;NOT
T1 &RJTABLE &TN /
T2 / /&CP   QN16 - Q16. Which of the following best
describes your age?
T3 TABLE &TN     QN16 - Q16. Which of the following
best describes your age?
R TOTAL RESPONDENTS; ALL
R Under 25;51-1
R 25 to 34;51-2
R 35 to 44;51-3
R 45 to 54;51-4
R 55 to 64;51-5
R 65 or over;51-6
R Don't Know/Refused;51-7
R NOT REPORTED;NOT
T1 &RJTABLE &TN /
T2 / /&CP QN21 - Q21. Respondent Sex
T3 TABLE &TN     QN21 - Q21. Respondent Sex
R TOTAL RESPONDENTS; ALL
R Male;57-1
R Female;57-2
R Not determined;57-3
R NOT REPORTED;NOT
```

Triple-S XML:

```
<?xml version="1.0"?>
<sss version="1.1">
  <date>18 FEB 2004</date>
  <time>15:55</time>
  <origin>CfMC MENTOR 7.7 v.20oct03 (1:16Feb04)</origin>
  <user>con to con</user>
  <survey>
  <!-- CfMC MESSAGE: please review all lines below which
  start with "CfMC MESSAGE" -->

       <title>EXAMPLE JOB</title>
       <record ident="A">
           <variable ident="1" type="single">
              <name>QN1</name>
              <label>Q1. How much do you agree with the
              following statement: The fast
              food at Road Runners is worth what I pay for
              it.</label>
       <position start="5" finish="5" />
       <values>
<!-- CfMC MESSAGE: values recoded from original format -->
       <value code="1">(5) Completely agree</value>
       <value code="2">(4) Somewhat agree</value>
       <value code="3">(3) Neither agree nor disagree</value>
       <value code="4">(2) Somewhat disagree</value>
       <value code="5">(1) Completely disagree</value>
       <value code="6">Don&apos;t Know/Refused to answer</value>
   </values>
</variable>

<variable ident="2" type="single">
   <name>QN2A</name>
   <label>Q2a. Please rate the following characteristics: The
   quality of the food.</label>
   <position start="6" finish="6" />
   <values>
<!-- CfMC MESSAGE: values recoded from original format -->
       <value code="1">(5) Very good</value>
       <value code="2">(4) Good</value>
       <value code="3">(3) Neither poor nor good</value>
       <value code="4">(2) Poor</value>
       <value code="5">(1) Very poor</value>
       <value code="6">Don&apos;t know/refused to answer</value>
</values>
</variable>
```

```
<variable ident="3" type="single">
   <name>QN2B</name>
   <label>Q2B. Please rate the following characteristics: The
   quality of service.</label>
   <position start="7" finish="7" />
   <values>
<!-- CfMC MESSAGE: values recoded from original format -->
       <value code="1">(5) Very good</value>
       <value code="2">(4) Good</value>
       <value code="3">(3) Neither poor nor good</value>
       <value code="4">(2) Poor</value>
       <value code="5">(1) Very poor</value>
       <value code="6">Don&apos;t know/refused to answer</value>
   </values>
</variable>

<variable ident="4" type="single">
   <name>QN5G</name>
   <label>Q5g. In response to the statement, &quot;Road Runners
   fast food provides good food at realistic and affordable
   rates.&quot; Do you . . .</label>
   <position start="8" finish="8" />
   <values>
<!-- CfMC MESSAGE: values recoded from original format -->
       <value code="1">(5) Completely agree</value>
       <value code="2">(4) Somewhat agree</value>
       <value code="3">(3) Neither agree nor disagree</value>
       <value code="4">(2) Somewhat disagree</value>
       <value code="5">(1) Completely disagree</value>
       <value code="6">Don&apos;t Know/Refused to answer</value>
   </values>
</variable>

<variable ident="5" type="single">
   <name>QN6</name>
   <label>Q6. Has anyone in your household participated in the
   entertainment available at Road Runners during the past three
   months?</label>
   <position start="9" finish="9" />
   <values>
<!-- CfMC MESSAGE: values recoded from original format -->
       <value code="1">Yes</value>
       <value code="2">No</value>
       <value code="3">Don&apos;t Know</value>
   </values>
</variable>
```

```
<variable ident="6" type="multiple">
   <name>QN6A</name>
   <label>Q6a. Which entertainment was participated in during the
   past three months?</label>
   <position start="10" finish="15" />
   <spread subfields="6" width="1" />
   <values>
      <value code="1">Video games</value>
      <value code="2">Billiards</value>
      <value code="3">Fun House</value>
      <value code="4">Musical Revue</value>
      <value code="5">Dunk the Moose</value>
      <value code="6">Other</value>
      <value code="7">Don&apos;t know/refused</value>
   </values>
</variable>

<variable ident="7" type="single">
   <name>QN6B</name>
   <label>Q6b. Have you used used the video games at Road Runners
   within the past three months?</label>
   <position start="16" finish="16" />
   <values>
<!-- CfMC MESSAGE: values recoded from original format -->
      <value code="1">Yes</value>
      <value code="2">No</value>
      <value code="3">Don&apos;t Know</value>
   </values>
</variable>

<variable ident="8" type="multiple">
   <name>QN8</name>
   <label>Q8. From your own experience and knowledge, what do you
   especially like about Road Runners?</label>
   <position start="17" finish="46" />
   <spread subfields="15" width="2" />
   <values>

      <value code="01">Good service/prompt service</value>
      <value code="02">Dependable/continuous service</value>
      <value code="03">The courteous employees they
      have/helpful</value>
      <value code="04">I like the food/good food</value>
      <value code="05">Food selection</value>
      <value code="06">Good prices</value>
      <value code="07">Computerized/accurate billing</value>
      <value code="08">Helpful in explaining billing
```

```
       questions</value>
       <value code="09">Good entertainment</value>
       <value code="10">Variety of entertainment</value>
       <value code="11">Nice family atmosphere</value>
       <value code="12">Established place/has been around for
       awhile</value>
       <value code="13">Accessible/Available/They&apos;re
       everywhere</value>
       <value code="14">Like everything/good place</value>
       <value code="15">Other</value>
       <value code="22">No problem/No complaints</value>
       <value code="23">Don&apos;t know/No answer</value>**
       <value code="24">Nothing</value>
     </values>
</variable>

<variable ident="9" type="single">
   <name>QN15</name>
   <label>Q15. What is the highest level of formal education
completed by the principal wage earner in your home?</label>
   <position start="47" finish="47" />
<values>
       <value code="1">Less than grade school</value>
       <value code="2">Grade school</value>
       <value code="3">Some high school</value>
       <value code="4">High school diploma</value>
       <value code="5">Technical school</value>
       <value code="6">Some college</value>
       <value code="7">College graduate</value>
       <value code="8">Graduate school</value>
       <value code="9">Refused</value>
   </values>
</variable>

<variable ident="10" type="quantity">
   <name>QN20</name>
   <label>Q20. How many children ages 18 and under are in living at
home?</label>
   <position start="48" finish="50" />
   <values>
<!-- CfMC MESSAGE: Asterisks in the data indicate a numeric
overflow -->
<range from="0" to="10" />
<!-- CfMC MESSAGE: numeric exception codes have been changed to
large numbers -->
<!-- CfMC MESSAGE: exception "RF" changed to "999" -->
<value code="999">RF</value>
```

```
    </values>
</variable>

<variable ident="11" type="single">
    <name>QN21</name>
    <label>Q21. Respondent Sex</label>
    <position start="51" finish="51" />
    <values>
        <value code="1">Male</value>
        <value code="2">Female</value>
        <value code="3">Not determined</value>
    </values>
</variable>

<variable ident="12" type="character">
    <name>DEMO1</name>
    <label>May I have your name so that my supervisor can verify
this information if necessary?</label>
    <position start="52" finish="86" />
    <size>35</size>
</variable>

<variable ident="13" type="single">
    <name>LAST</name>
    <label>STATE CODE</label>
    <position start="87" finish="88" />
    <values>
<!-- CfMC MESSAGE: values recoded from original format -->
        <value code="01">Alaska</value>
        <value code="02">Alabama</value>
        <value code="03">Arkansas</value>
        <value code="04">Arizona</value>
        <value code="05">California</value>
        <value code="06">Colorado</value>
        <value code="07">Connecticut</value>
        <value code="08">District of Columbia</value>
        <value code="09">Delaware</value>
        <value code="10">Florida</value>
        <value code="11">Georgia</value>
        <value code="12">Hawaii</value>
        <value code="13">Idaho</value>
        <value code="14">Illinois</value>
        <value code="15">Indiana</value>
        <value code="16">Iowa</value>
        <value code="17">Kansas</value>
        <value code="18">Kentucky</value>
        <value code="19">Louisiana</value>
        <value code="20">Massachusetts</value>
```

```
             <value code="21">Maryland</value>
             <value code="22">Maine</value>
             <value code="23">Michigan</value>
             <value code="24">Minnesota</value>
             <value code="25">Missouri</value>
             <value code="26">Mississippi</value>
             <value code="27">Montana</value>
             <value code="28">North Carolina</value>
             <value code="29">North Dakotaa</value>
             <value code="30">Nebraska</value>
             <value code="31">New Hampshire</value>
             <value code="32">New Jersey</value>
             <value code="33">New Mexico</value>
             <value code="34">Nevada</value>
             <value code="35">New York</value>
             <value code="36">Ohio</value>
             <value code="37">Oklahoma</value>
             <value code="38">Oregon</value>
             <value code="39">Pennsylvania</value>
             <value code="40">Rhode Island</value>
             <value code="41">South Carolina</value>
             <value code="42">South Dakota</value>
             <value code="43">Tennessee</value>
             <value code="44">Texas</value>
             <value code="45">Utah</value>
             <value code="46">Virginia</value>
             <value code="47">Vermont</value>
             <value code="48">Washington</value>
             <value code="49">Wisconsin</value>
             <value code="50">West Virgina</value>
             <value code="51">Wyoming</value>
         </values>
      </variable>
    </record>
  </survey>
</sss>
```

APPENDIX G: HARDWARE/SYSTEM

G

RECOMMENDED HARDWARE

Minimum Requirements

STANDALONE (DOS)

1 To compose questionnaires:

Program: Script Composer/Prepare

Memory: 128 megabytes RAM

Disk Drive: (1) 8 gigabyte hard disk

(1) 1.4 megabyte floppy drive or CD rom

Windows 95/95 or greater

CPU: 1.0 GHZ Pentium

Ports: Serial, parallel, or USB port must have CfMC security key installed

2 Interviewing Stations:

Program: Survent

Memory: 640 Kbytes RAM

Disk Drive: 1 80 Mbyte+ hard disk recommended

CPU: 256 MHZ IBM 486+

SUPERVISED (DOS)

One computer (not the network server) must be available for CfMC dialup support and remote diagnostics, with a 56K modem and an outside phone line.

A networked Survent operation requires:

- 1 Novell Server - for managing the processing of messages in the network

- 1 STDYSRVR computer- to keep the questionnaire files, phones, completed interviews
- 1 Supervisor computer- some like 1 supervisor for every 10-20 interview stations
- n Interviewing stations

Certification by CNE (Novell Certified Network Engineer) if Novell

• Novell/Windows NT+ File Server

Pentium IV 1.7GHz, 512MB memory,

20GB hard disk

• CfMC STDYSRVR Pentium IV 2.0GHz, 512MB memory,

20GB hard disk, MS-DOS 6.22

Windows 95+

• n Interviewing PCs, 1 Supervisor (min.)

20GB hard disk, MS-DOS 6.22

Windows 96+

• Network cards (1 per machine)-Ethernet

- 3 active ports (1 for every 7 connections)
- One PC needs modem or internet access for support from CfMC
- Uninterruptible Power Source (UPS)
- Network cabling
- Tape backup
- Windows NT+ or Novell Netware 4.1 or higher (25 user)
- Remote access software (e.g., Carbon Copy, PCanywhere or PROCOM +)

Hardware and System Checklist Discussion

With your disk(s) or CD comes a cover sheet that tells you how to install the software. In addition to copying the files onto your hard drive, networked users need to do some additional steps, depending on your hardware platform. The three hardware platforms are discussed separately below, but the process you follow is basically the same. Non-DOS users should read the DOS section in addition to their own section. Standalone DOS and UNIX users should read the DOS section and see the special note at the end of that section that pertains to standalone use.

DOS PC NETWORK CHECKLIST

Here are some factors to keep in mind when preparing for the use of networked Survent:

1 **NETWORK VERSION:** Novell Netware 2.11 or later. Windows NT+ also supported

2 **TOPOLOGY:** Ethernet. It is recommended that the CfMC SERVER computer have a direct connection to the network server; interviewing stations may share a hub.

3 **CABLES:** Cables must be terminated with a machine or 93 ohm resistor. All cables must be at least 2 feet away from fluorescent lights.

4 **POWER:** The Network file server and CfMC data server must both be connected to a UPS (uninterruptible power supply).

5 **MEMORY:** All machines must have at least 28mb memory available to run the network and CfMC's Survent concurrently.

6 **DISK DRIVES:** It is recommended that all computers have disk drives. Although it is possible to run Survent with all interviewers writing to the network, it is much more efficient if local files are written to a disk drive. Disk drive should be 10 MG minimum.

7 **FILES:** The CfMC data server must be able to open enough files to operate the system.

This is set in the CONFIG.SYS file for that computer or the "FILE HANDLES=" command in your network configuration file (either SHELL.CFG in older versions of Novell or NET.CFG in newer versions). The formula for number of files open is:

15 + (# of studies run concurrently * 8) + (max # of stations operating)

This means "FILES=100" will handle 50 stations running up to 4 studies. The supervisor stations must have "FILES=" set to:

15 + (# of studies under supervisor * 3) + (# of stations under supervisor)

Note: The maximum number of files open in early versions of Novell is 256. You may test the number of files you open by using the CfMC program TESTOPEN at the DOS prompt from the computer in question.

Setting up the Network

Throughout this section the network drive designation is referred to as the F: drive, and the software is assumed to be in F:\CFMC. Study files are assumed to be in the directory designated by the $HOME variable. This $HOME variable can also be F:\CFMC, or it can be something totally different.

Your interviewing files do not have to live in the same directory as your CfMC program files. Substitute the correct letter drive and directory designations to match how your shop network is set up.

Network User Setup

The Novell network manager should use SYSCON (or whatever network administration program your system uses) to create three users called: CFMCSUPR, MONITOR, and INTERVIEWER. CFMCSUPR should be assigned full access rights to directory F:\CFMC and F:\$HOME. MONITOR and INTERVIEWER should have read/write access to the $HOME directory, with ead/execute access to the CFMC directory.

Network Directory and File Setup

Make the following directories on the network file server (usually the F: drive) using the MKDIR (MD) command. You must have parental rights to the F:\CFMC directory to do this, as described earlier:

(and remember that the $HOME directory is an actual named directory; see the discussion above)

\CFMC

\CFMC\CONTROL

\CFMC\SURVENT

\CFMC\GO

```
\CFMC\SUPPORT
\$HOME\FONE
\$HOME\DATA
\$HOME\QFF
\$HOME\QUOTA
\$HOME\RESUME
\$HOME\STUDY\<study names> (optional)
\$HOME\IPCFILES
```

Here is a description of the purpose of each directory:

Directory	**Description**
CFMC	Base directory for CfMC software.
CONTROL	Contains the MSGFILE, TTYFILE, INITIAL, ZONETABL and other system control files.
GO	Contains all executable (EXE) and batch (BAT) files.
SUPPORT	Contains files used by the CfMC Utilities.
SURVENT	Contains example questionnaires displaying various features or supporting operations.
$HOME	Home directory in which active study files are stored.
DATA	Contains study data (TR) files here for all studies.
FONE	Contains study phone (FON) and phone index (FNX) files.
IPCFILES	Contains communications files (one IPC file for each device), the employee information file (EMPLOYEE.XXX) and system configuration files (STATIONx.CFG, STUDIEx.CFG, and DASHSTDx.CFG).
QFF	Contains the questionnaire (QFF) files for all active studies.
QUOTA	Contains the quota (QUO) files for all the active studies.
RESUME	Contains data from suspended interviews that may be resumed later. Each study’s resume files are stored in a subdirectory named <study>.R_.
STUDY\<study>	Where <study> is the name of any study. This subdirectory would be used to store the specification and supporting files for a particular study.

Copying Files from CfMC Install Diskettes

Your install disks should load the files listed here (among others) in the following subdirectories of CFMC or $HOME:

File Name Directory Program description

File name	Directory	Program description
MAKECFG.EXE	GO	Builds or modifies STATIONS, STUDIES, and DASHSTUD.CFG files
STDYSRVR.EXE	GO	Data/Phone/Quota file server program
SURVENT.EXE	GO	Interviewing program
SURVMON.EXE	GO	Monitoring program
SURVSUPR.EXE	GO	Supervisor program
FONEBULD.EXE	GO	Builds phone files
FONESIM.EXE	GO	Phone system simulation program for testing
FONEUTIL.EXE	GO	Modifies or reports on phone files
CLEARIPC.BAT	GO	Deletes and rebuilds all communication files
CONFIG.BAT	GO	Runs MAKECFG program
MONITOR.BAT	GO	Runs SURVMON program
PHONERPT.BAT	GO	Runs PHONERPT utility for phone reports
SERVER.BAT	GO	Runs STDYSRVR program
START.BAT	GO	Runs SURVENT program
SUPER.BAT	GO	Runs SURVSUPR program
MSGFILE	CONTROL	Messages used by CFMC programs
PHONEDEF	CONTROL	Phone definitions used in FONEUTIL and SURVSUPR
TTYEXAM	CONTROL	Example TTYFILE
ZONETABL	CONTROL	Used by FONEBULD to check for proper time zone information
EMPLEXAM.XXX	IPCFILES	Example Employee information file
PHRPT.DB	SUPPORT	Database file used by PHONERPT utility
PHRPT.QFF	SUPPORT	Phone report utility questionnaire
PHRPT.SPX	SUPPORT	Spec file version of phone report utility
ZONETABL.DOC	SURVENT	ASCII version of ZONETABL

File name	Directory	Program description
AUTOD.QPX	SURVENT	Example spec file for doing auto-dialing using a modem
EXAM1.QPX	SURVENT	Example spec file for testing
EXAM2.QPX	SURVENT	Example spec file for using phone system
NUMBERS.RAW	SURVENT	Sample raw phone number file

Follow any special instructions given on the cover sheet that came with the disks, but typically you install by doing the following:

```
    A:INSTALL A: F:
```

where A: is the floppy drive where you inserted the installation disk, and F: is the network drive where you want the software to be put. If you specify a directory after the F: (eg., F:ACME), you are asking for the software to be installed within that (ACME) directory.

Initializing the CfMC System Files

Before running CfMC programs for the first time, you must edit the PARMFILE and INITIAL files in the CONTROL directory, and the EMPLOYEE.XXX file in the IPCFILES directory to provide interviewer IDs for the interviewers. You will also need to build the system configuration files in the IPCFILES directory.

The INITIAL file is read by every CfMC program that has a Spec File or List File prompt when starting up. CfMC provides the file INITEXAM which is an example of startup parameters you may use.

Rename it or copy it to INITIAL. For the most part, there are no requirements for this file other than that the lines ">DEFINE @DOS" and ">DEFINE @WATCOM" are there. Talk to CfMC about other potential startup parameters.

The PARMFILE is required for the programs to run. CfMC provides the file PARMEXAM that you can simply rename or copy to PARMFILE. The only information that is required is the line "TIMEZONE:xx" where xx is the number of your timezone (with 1 being Greenwich Time, 5 being eastern time in North America, and 8 being pacific time). You will also need to add the

EXPIRATION: and VALIDATION: strings provided by CfMC that allows you to run the software. See *4.4.4 INTERVIEWING WITH SURVENT* for more information.

The EMPLOYEE.XXX file in the IPCFILES directory provides interviewer information, such as IDs, names, special options, and other user supplied information. Use the file EMPLEXAM.XXX as an example. See *4.4.4 INTERVIEWING WITH SURVENT* for more information. Add as many IDs as you wish and save the file. You are now done editing CfMC system files.

Now you need to build the CfMC server stations and studies files before starting the CfMC server. To do this, enter "CONFIG MAKENEWFILES". This builds the STUDIES.CFG, STATIONS.CFG and DASHSTD.CFG files in the CFMC\IPCFILES directory. This is where it keeps track of which stations and studies are running under the server.

Setting up the CfMC Server Station

The CfMC Server computer controls all access to study files and communications between all devices running under the CfMC system. However, it is NOT a "file server" in the sense of the Novell file server. It is simply a program that runs on the network that routes all messages between CfMC-controlled computers and updates all files for the studies running under it.

The requirements for the CfMC server are greater than for the other computers on the system. In particular, the number of files you can open must be set high, the amount of environment space should be high, and the server should have read/write access to all of the relevant directories. If there is a choice of computers for various operations, the CfMC server computer should be given priority regarding CPU power and disk I/O.

NOTE: Disk space is not an issue since all files are stored on the network server. The CfMC server should run under the user CFMCSUPR like the supervisors, or some other user set up with the same access rights.

Setting the Maximum Number of Files that May Be Opened

The CfMC server needs to open many files. To set this, use either the "FILES=" parameter in the CONFIG.SYS file, or the "FILE HANDLES" setting in the NOVELL network.

Change the FILES= in the CONFIG.SYS of the machines that will be running the SERVER to a number > 100 (Under Windows NT, edit the file CONFIG.NT in the \WINNT\SYSTEM32 directory).

The formula for the FILES= setting is:

15 + (# of studies running * 8) + (# of stations running)

To set this at the network level, create or edit the file NET.CFG in the directory from where you login to the network on the SERVER machine. Add the line "FILE HANDLES = 200" to set the files to 200.

Setting CfMC System Variables

Like all of the computers, the server will need a search path to directory F:\CFMC\GO, and the "CFMC" variable set. This may be done by using SYSCON to update the login script, by using the DOS SET command in the AUTOEXEC.BAT file (or \WINNT\SYSTEM32\AUTOEXEC.NT under WINDOWS NT), or by using the CfMC-supplied SERVER.BAT batch file.

To do this in the login script using the Novell program SYSCON, use the commands:

```
MAP S1:=F:\CFMC\GO
DOS SET CFMC="F:\\CFMC\\"
```

To do this with the DOS SET commands (such as in AUTOEXEC.BAT) enter:

```
SET PATH=F:\CFMC\GO;%PATH
SET CFMC=F:\CFMC\
```

NOTE: If the CFMC variable is not set, the system displays error messages when you attempt to start CfMC programs.

The CfMC server must also have directory references to all of the study directories where files are accessed or stored as follows:

```
SET CFMC=F:\CFMC\
SET CFMCRESUME=F:\$HOME\RESUME\
SET CFMCQFL=F:\$HOME\QFF\
SET CFMCDATA=F:\$HOME\DATA\
SET CFMCFONE=F:\$HOME\FONE\
SET CFMCQUOTA=F:\$HOME\QUOTA\
SET CFMCCFG=F:\$HOME\IPCFILES\
```

Increasing Environment Space

You should also add environment space by using the SHELL= command in the CONFIG.SYS (or CONFIG.NT) file located in the root directory of the local startup drive (or \WINNT\SYSTEM32).

To do this, at the top of the CONFIG.SYS file add the line:

```
SHELL=COMMAND.COM /E:1024 /P
```

This should be done for SERVER machine, SUPERVISOR machines, and MONITOR machines.

If you are running out of environment space when running inside of a DOS box from Windows you can increase the environment space by doing the following:

8 Edit the file SYSTEM.INI in the Windows (or WINNT) directory.

9 Find the line that says "[NonWindowsApp]" and add the following line to what might already be there: "CommandEnvSize=1024".

This will set the environment space to 1024 characters. This is usually more than enough to do whatever you need.

Testing the Server

To test the CfMC server, boot up the computer and enter "SERVER <ldev>", where <ldev> is the assigned device number. This should execute the file F:\CFMC\GO\SERVER.BAT. If the program operates successfully, you will see messages about opening the communications files, then the program will display dots or a wheel of characters indicating that it is running.

Press **Ctrl-Y** to stop the server from dotting and allow you to issue commands. For our test, type the command "TESTOPEN". The program will display numbers and open as many files as it can before aborting. ***WARNING:*** Do not run the process TESTOPEN while you have an active study running as this process will cause the study server to abort. The last line should read: “Done. File=-1 nf=<#>.” <#> will be the number of files that may be opened. Check that this is high enough for your CfMC system using the formula:

15 + (# of studies running * 8) + (# of stations running)

In order to run 30 interviewers on 5 different studies at a particular time, this number should be at least 85. If you cannot open enough files, check the "FILES=" setting in the CONFIG.SYS file, or the "FILE HANDLES=" setting in your network configuration file (either SHELL.CFG or NET.CFG in the login directory). Modify the settings, reboot the computer, and test again if necessary.

Setting Up Supervisor and Monitor Stations

You may run as many study Supervisors or Monitors as you like with the CfMC system. Each Supervisor station will start one or more interviewer stations. Monitors may monitor any supervised or shared files interviewer on the system (if you do not password protect the study the interviewer(s) are running).

Like the CfMC server station, you must set the PATH to point to the F:\CFMC\GO directory and also set the CFMC variable. You should also use the same procedures to add environment space and set the allowable open files to match the CfMC SERVER.

Before starting the programs, enter "TESTOPEN". You will see a list of numbers, the last of which is the total number of files the station may open. Adjust the FILES= or FILE HANDLES= parameters until this is high enough.

To set the search path and other definitions, you may use the CfMC-supplied SUPER.BAT and MONITOR.BAT batch files (which will automatically set the path and definitions for you) or you can set the following variables in the AUTOEXEC.BAT or your CFMCSUPR login script.

```
SET CFMCRESUME=F:\$HOME\RESUME\
SET CFMCQFF=F:\$HOME\QFF\
SET CFMCDATA=F:\$HOME\DATA\
SET CFMCFONE=F:\$HOME\FONE\
SET CFMCQUOTA=F:\$HOME\QUOTA\
SET CFMCCFG=F:\$HOME\IPCFILES\
      SET CFMC=F:\$HOME\
```

To start the supervisor, enter "SUPER". To use an interviewer monitoring station, use the “MONITOR” command. Station numbers will need to be used or assigned in the BAT files.

Setting Up the Interviewer Stations

Like the other stations, you should set the CFMC path and variables needed to access relevant files, set the environment space and maximum number of files. Interviewers need only open the default of 32 files.

Each interviewer must have a search path to directory F:\CFMC\GO. This search path may be created by adding commands to the user's login script or by using the DOS SET command. To set the search path and other definitions you may use the CfMC-supplied START.BAT batch file (which will automatically set the path and definitions for you) or set it in the login script or AUTOEXEC.BAT file (see above for details).

Interviewers must have Read-Write access to the $HOME directory.

Interviewers may interview from anywhere on the network drive except the root directory of the drive (which only allows 255 files to be saved), but would be best off running from a local hard drive. A typical directory for interviewing would be the C:\INTVWR directory. You may make sure interviewers run in that directory by uncommenting the following 2 lines in the file START.BAT:

Example:

```
      C:
      CD \INTVWR
```

Having the interviewers run on the C: drive is recommended over running off of the F: (network) drive. This reduces the load on the network greatly.

Testing the Survent Interviewing System

Logon as CFMCSUPR on the network. Move to the directory \$HOME\STUDY. Copy \CFMC\SURVENT\EXAM1.QPX to subdirectory \$HOME\STUDY\EXAM1 (you may need to create this subdirectory).

Compiling the Questionnaire and Keys

In order to compile a questionnaire, you must have a CfMC PREPARE hardware key attached to the back of the computer, and the SENTINAL key driver installed. Instructions for this should come with your installation disk(s).

The key must be put into a parallel or USB port depending on the key type; parallel ports (sometimes called printer ports) are female. Insert the male end of the key into the female end of the parallel port. Seat it firmly as noted above.

Compile the specification file, EXAM1.QPX, by entering the following command:

```
    PREPARE EXAM1.QPX -EXAM1.LST
```

When the compile is finished, enter the command LOADSTDY EXAM1 to copy the resulting files to their proper CfMC directories.

At the CfMC server station, enter "SERVER" or, if you have not set the environment variable STATION, enter "SERVER <ldev>" where <ldev> is the station's ID number. <ldev> must be an integer number between 1 and 255.

Once the server is running, you should not have to do anything to this machine until you want to shut it down. You will see dots indicating it is running.

On the machine designated for being the Supervisor login as CFMCSUPR. Enter "SUPER" or, if you have not set the environment variable STATION, enter "SUPER <ldev>" where <ldev> is the station's ID number. <ldev> must be an integer number between 1 and 600, and must be a different number than that used for either the SERVER or any of the interviewing stations.

Once SURVSUPR is running, press **Ctrl-Y**. You will be presented with the Supervisor's menu.

Entering "HELP" will print this menu if it goes away.

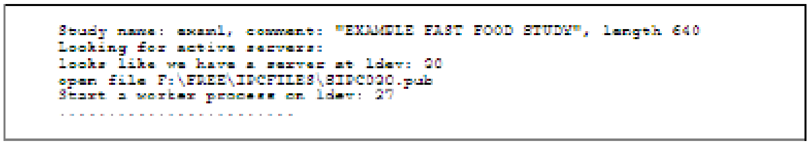
Enter "STS <interviewing station’s number> EXAM1"

Example:

```
STS 12 EXAM1 (start station 12 on study EXAM1)
STS 12-20 EXAM1 (start stations 12-20 on study
EXAM1)
```

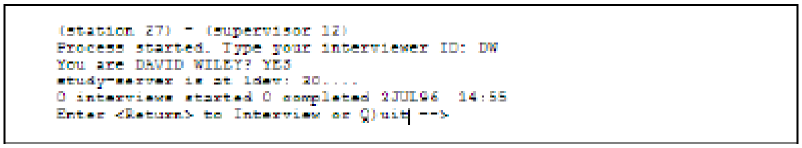
Note: If you are not sure what the interviewing station's number is, go to that machine and enter START at the operating system prompt. The station number will print then the program will abort.

You should now see something like this on the SURVSUPR screen:



```
Study name: exam1, comment: "EXAMPLE FAST FOOD STUDY", length 640
Looking for active servers:
looks like we have a server at ldev: 20
open file F:\FREE\IPCFILES\SIPC020.pub
Start a worker process on ldev: 27
.........................
```

You are now ready to start interviewing. On the machine(s) designated for interviewing, login as INTERVIEWER. Enter "START" or, if you have not set the environment variable STATION, enter "START <ldev>" where <ldev> is the station's ID number. <ldev> must be an integer number between 1 and 9999, and must be set to a different number than that used for the Server or Supervisor. If the system is working properly and the device numbers match, you will see the following:



```
(station 27) - (supervisor 12)
Process started. Type your interviewer ID: DW
You are DAVID WILEY? YES
study-server is at ldev: 20....
0 interviews started 0 completed 2JUL96  14:55
Enter <Return> to Interview or Q)uit -->
```

Press **Return** to run the questionnaire EXAM1. When you are done with the interview, you should see a message about the case ID assigned to the data case saved, then the current time and number of interviews started and completed will display, and you will be prompted to begin another interview.

Testing the Survent Phone Sample Control System

To test the phone system, copy the file \CFMC\SURVENT\EXAM2.QPX and NUMBERS.RAW to the subdirectory \$HOME\STUDY\EXAM2 (which will need to be created first).

Run the program FONEBULD and say "SHOPFILE". You will see a screen of default phone system control parameters. Set at least the local time zone and the time zones you would be interviewing in (usually 5, 6, 7, and 8 if in North America). Set any other parameters you know like when your interviewing shop will be open every day. Press **ESC** and enter "WRITESHOP \CFMC\CONTROL\SHOPFILE" to save these default settings for the next time you use the phone system.

Enter "**GO**" and you will be prompted for a study code and file of numbers to read. In this case, the study code should be "EXAM2" and the file of numbers "NUMBERS.RAW". Press **Enter** twice and the numbers will be processed. Then copy files as follows:

```
COPY EXAM2.QFF F:\$HOME\QFF
COPY EXAM2.QUO F:\$HOME\QUOTA
COPY EXAM2.FON F:\$HOME\FONE
COPY EXAM2.FNX F:\$HOME\FONE
```

Or enter the command “LOADSTDY EXAM2” to copy the files automatically. Use the same procedures as for EXAM1 to start interviewers. When you get into the interview, you will see a phone number come up and be asked to assign a status to the number. Set some statuses and complete some interviews. Use the Supervisor command "SPI EXAM2" to see how many numbers you've called and how the numbers are getting re-scheduled. You may also run the program FONEUTIL to get information on the current status of the phone file or make changes to it.

Backing Up Your Data

To do a backup of your data on a daily basis, someone should create and run a BAT file on a daily basis that copies data to another directory. Tape backup should be performed regularly to assure that data is not mistakenly deleted or somehow lost in case of a network failure.

STANDALONE USE

To run the Standalone version of Survent, you will install your software as noted above (except you will probably be putting it on your C: or D: drive). You will need to modify the PARMFILE to make sure that the correct time zone is stated if you will be using the phone system, and update the INITIAL file if desired. There is no need for an EMPLOYEE.XXX file, as it is not used in standalone mode.

Make sure that your path includes where you have installed the software (typically, by adding C:\CFMC\GO to your PATH statement in your AUTOEXEC.BAT file). Also make sure that the CFMC and CFMCDRIVE environment variables are set (typically also in your AUTOEXEC.BAT file by including lines such as SET CFMC=C:\CFMC\ and SET CFMCDRIVE=C:).

All of your files will be in the same directory, so there is no need to create the extra subdirectories off of CFMC. Where you do the interviewing is up to you. You may choose to work out of \$HOME\STUDY, but there is no need to do so. Just make sure that you are not interviewing in any of the other directories within CFMC that got created by the installation (so if you mess things up, you will have your study files stored separately from the program files). You will still need to use the key to compile questionnaires, as noted above.

Environment space should be increased as noted above, but only to 512. See Chapter 4 for information on other interviewing differences.

UNIX

We will use SCO UNIX as an example, but all the UNIX systems have a similar install procedure.

Log on the system as "root" to provide full capabilities. Then, use the "scoadmin" program to create tne user named "cfmc" and another named "intv001". The user logins should boot up in the C Shell (csh).

The number of files and/or processes allowed must be set high enough to handle all interviewers once they are logged on (modify the Kernel parameters to set this). See Kernel update options.

Then untar the install command from the installation disk per the instructions; create a directory to put the CFMC software in, copy the install file to that directory, and type "./install". The directories and permissions below will be created by the install file (/cfmc/ would be the name you used for the cfmc data area):

```
/cfmc/go           r-xr-xrwx
/cfmc/support      r--r--rw-
/cfmc/control      r--r--rw-
/cfmc/qff          r--r--rw-
/cfmc/quo          rw-rw-rw-
/cfmc/ipcfiles     rw-rw-rw-
/cfmc/data         rw-rw-rw-
/cfmc/fone         rw-rw-rw-
/cfmc/resume       rw-rw-rw-
```

If you need to modify permissions, use "chmod +rwx <directory name>".

These environment variables should also be set (with the default definitions, usually done in the .login file or in the command shell script which controls the process with a "setenv CFMC /cfmc/" kind of command) or in the .login script:

```
setenv CFMC       /cfmc/
setenv CFMCRESUME /cfmc/resume/
setenv CFMCCFG    /cfmc/cfg/
setenv CFMCFONE   /cfmc/fone/
setenv CFMCDATA   /cfmc/data/
setenv CFMCQUOTA  /cfmc/quota/
setenv CFMCQFL    /cfmc/qff/
```

The CfMC software must be run under the C shell of UNIX. Standard login files are provided in the "control" directory; login.exam is an example .login file for the CfMC user; cshrc.exam is the standard .cshrc file. Both of these files must be placed in the login directory of each CfMC user.

Interviewers should login under a different user-name than CFMC, such as intv001. A separate user name should be assigned for each interviewing station so supervisors can assign the same device numbers regardless of who is working there. For system security reasons, CfMC recommends that interviewers be allowed only to interview, with no access to the system other than through the CfMC Survent program interface. A suggested interviewer .login file ends with:

```
setenv CFMC /cfmc/
setenv CFMCRESUME /cfmc/resume/
setenv CFMCCFG /cfmc/cfg/
setenv CFMCQFL /cfmc/qff/
/cfmc/go/survent "supervised" "con" "con" "ldev:$ldev"
exit
```

This .login sets the requisite parameters, executes Survent, and exits back to the login prompt.

Under UNIX, you can set the terminal info using standard login variables (preferred if you are logging in with telnet sessions) or the TTYINFO file specifies each terminal's terminal type and assigns each a specific number if you have hard-wired stations. See the example /cfmc/control/login.exam to see how to set the variables. The allowable terminal types are listed in Chapter 4 of the manual. The terminal name may be found by logging on to a station and using the command "who am i" to see a list of device names. The format of the TTYINFO file is:

```
<device type> <device name> <device number> <time zone>
<extension>
ansi tty01 01
ansi tty02 02
wyse50 ttypa01 03
```

See *4.4.4 INTERVIEWING WITH SURVENT, The Ttyinfo file* for more information.

This file should be sorted by device name. If you are using a sound server or dialer, you must have the sound/dialer channel specified. Once you are done editing the TTYINFO file, add the interviewer names to the "employee.xxx" file in the /cfmc/ipcfiles directory, and check the "/cfmc/control/parmfile" file settings.

First you must start a server. Type “server” to run from a station, or “server bg” to run the server in the background. This will make sure no other servers are running, create a new stations and studies file, and start the “snapper” program to keep track of interviewer activity before actually starting the CfMC server program. The server will run in the /cfmc/super directory by default and write server log files there. Server output information will be written to the file servlog in the /cfmc/ipcfiles directory.

If you have standard numbers assigned to each station, when you run the Supervisor or interviewers, simply enter ‘super’, or ‘start’. There is no need to designate a station number; the system automatically picks it up from the TTYINFO file or from the LDEV variable setting in the .login script. If you have not preset the device number in some way, specify the device number after the command.

To use a menu of CfMC programs, enter "cfmcmenu". This includes easy access to the programs (by number), and a help file with descriptions of each program.

To print files, use the CfMC UNIX utility "cprint", or the "lp" command.

You should also familiarize yourself with the following UNIX programs/commands:

Program/ command	**Description**
man	read the manual description of various commands
vi	standard UNIX file editor
pg	page through files (without editing them)
kil	kills unwanted sessions (use "kil -9 <session number>" to kill a process unconditionally).
who	to see which sessions are running
l	list files and directories (or use the CfMC supplied "dir" command)
lp	to print files
mv	move files from one place to another

Program/ command	Description
rm	remove files
copy	copy files from one place to another
ps	show processes; use ps -ef to see all information on all processes
tar	make an archive tape or diskette or read files off the same.
doscp	copy files to/from a DOS subsystem (such as a diskette)
dosdir	et a directory listing of a DOS subsystem

Device Number Assignment

In UNIX you can use device numbers 1-2000.

In DOS, you can use numbers 1-256.

SNAPALL Program

The SNAPALL program facilitates multiple CfMC server processes. The program also allows CfMC servers to write to their local $CFMCCFG directory instead of having a globally accessible /cfmc/cfg directory.

Snapall writes to file "/cfmc/cfg/timestamp.lst" which has counts of current processes. It also uses the file /cfmc/cfg/accounts.lst", which has a list of the various accounts files (which makecfg creates) to check.

For example:

```
/home/smcr/rel80/cfg/accounts.cfg
/home/smcr/demo8/cfg/accounts.cfg
/home/smcr/rel77/cfg/accounts.cfg
/home/joe/rel80/cfg/accounts.cfg
```

Other CfMC programs read from timestamp.lst to make sure it has current info (the snapper is running) and to get a count of the available processes, which are the max for each type from the CfMC control string minus the sum of that type from the various accounts.cfg files.

Note that multiple CfMC versions can be used. In version 7.7, makecfg looks to see if there is a timestamp.lst and accounts.lst file to decide whether it should be working with the file "${CFMCCFG}stat77.cfg". If not, it acts on the file "/cfmc/cfg/stat77.cfg" as before.

APPENDIX Y: PREPARE SYNTAX

Y

REFERENCE GUIDE

This REFERENCE GUIDE provides a summary of Survent statements in chapter order; for a more detailed explanation see the section references.

STUDY HEADER: (2.3.2 Study Header Statement)

```
[study name,options]
[study name,case length,"study comment",options]
```

Study name: 3 to 8 alphanumeric characters, starting with a letter.

Options:

(Ampersand (&) Used to continue the study header to the next line.)

Interviewer Display and Control Options

Statement	Explanation
ALLOW_ABORT	Allows the interviewer to abort an interview by typing ABORT.
ECHO_CATS=option	ECHO_CATS=optioncode highlighted) when the interviewer presses **Enter**; an extra **Enter** is required to move to the next question. The option can be SINGLE, MULTI or ALL.
ERROR_BEEP	Alerts interviewer with a beep when an invalid response is encountered; an error message also appears on the screen.
FLAG_DISALLOWED_CATS	Shows all responses in the response list for CAT and FLD question subtypes C and D and I, but highlights any that are not allowed as responses.

Statement	Explanation
HIGHLIGHTCATS_MATCH_TEXT	When using HIGHLIGHT_CATS mode on !CAT or !FLD questions to use an interactive screen on a code list, this says to match text the respondent enters to the TEXT of the responses rather than the response CODES. Use subtype “M” to do this on a particular question.
INFO_BETWEEN=#	Specifies what to print on the screen between interviews. You can show the # starts/completes, time since last interview/start of session, interviewer ID and Idev #, and phone information.

LANGUAGE=	Controls what languages are used in multi-language interviews. If you have not specified a language here, you cannot have \Lxx language sections in the questionnaire that only display when that language is set.
	Example:
	`Language=(SET=9aa bb cc)`
	`CHARACTERSET=type`
	`speaking=aa)`
	Languages can be any two-character code you want, but specific codes are supported for CfMC error messages (eg. FR for French). Use set=(F=Fr) to allow 1 character \L coding for French. CharacterSets can be Universal, Extended_ASCII, Shift_Jis, Multi_byte, and ASCII.
	Speaking= controls the current language, the default is the first on the language list.

SHOW_LAST_RESPONSE	This shows the response of the previous question at the bottom of the screen of each question.
SHOW_QUESTION_LABELS	Prints question labels in the lower right corner of the screen during an interview. SHOW_QUESTION_LABELS=UR will place the question label in the upper right corner of the screen.

Statement	Explanation
TEXT_HELP	Displays "F1 for help, ESC to exit" in the bottom line of a TEX question response box (DOS, UNIX only).
TIME_ON_SCREEN=<LINE#>	Shows the time and time since the start of the interviewing session in minutes, right- justified on the line specified.

Survent Operational Options

Statement	Explanation
-AUTOMATIC_CASEID_INCREMENT	Says to not have Survent utomatically provide case IDs for the data file. Interviewers will be prompted to assign an ID.
CARD_ID_FORMAT	Puts the case ID on every 80-column record of the file and a record ID in the location specified here.
CASE_ID=col.wid	Places the case ID in the specified column location, with the specified width.
COMMENT="text"	Allows a comment which will display when an interview is started.
FAILED_RESUME=option	Determines how to handle changed questionnaires when they are resumed and the changes cause the new and old version not to match. The option can be BLOW (stop the interview), STARTHERE (start where the first difference occurs), or RESTART (at the beginning).
INTERVIEWER_LOGGING	Causes interviewers respones on that study to be logged (default). Specifying Log=After_Every_Question will save the log file immediately after every question, so that if you are trying to find a problem (where the program is aborting abnormally for some reason), you can save responses up to that point and try to reproduce the problem. Log=No or –Log turns logging off for this study.

Statement	Explanation
NEXT_CASE_ID=#	Sets the initial case ID in the data and quota files, or sets the next ID if the data file already exists.
ONE_INTERVIEW	Survent to start the interview without the standard interviewer prompt and to exit after one interview is completed.
QUOTA_CRC_CHECK	Controls whether to check the quota file for a valid names block. Say -QUOTA_CRC_CHECK to override this if you have one questionnaire that had quota updates and others running at the same time with the same study name that didn't.
REDO_SPECIAL_BLOCK_ON_RESUME	Controls whether to re-execute the “SPECIAL” block when resuming the interview, default is yes, use – keyword to not reexecute it.
TEST_ONLY	Creates questionnaires that can be tested by Survent and used for REFORMAT or other programs, but do not generate data. Allows you to compile questionnaires in DOS without requiring a security dongle connected to the computer.
TIME_QUESTIONS	This will record the time between questions in the server log file.
TIMESTAMP	Requires that the QUO file built or updated during this compile be used for interviewing.
USE_DUMMY_IVR	Required if the questionnaire will be using the “Dummy interviewer” feature to post-process phone numbers returned from a predictive dialer.

Supervisor/Monitor Related Options

Statement	Explanation
ALLOW_VIEW	Allows View mode without requiring knowledge of the study password.
COUNT_AS_COMPLETE= #	Allows you to specify up to 5 statuses to be counted as completes in the SUPERVISOR DAI menus.
DASH_BOARD	Places the case ID in the specified column location, with the specified width.
MODIFY_CASE_ID=Y/N/P	Specifies the supervisor's access to modifying the "next" case ID (yes, no, or with a password).
MODIFY_FONE_FILE=Y/N/P	Specifies the supervisor's access to modifying the FON file parameters. (yes, no, or with a password).
MODIFY_QUOTA_FILE=Y/N/P	Specifies the supervisor's access to modifying the quota file (yes, no, or with a password).
-MONITOR	Disallows monitoring from SURVMON.
PASSWORD=pass	Allows you to assign a password to access View mode in Survent, the MPF, QSS MOD, and MODID commands in SURVSUPR, and monitoring in SURVMON.
TRIPLE_QUOTAS	Creates three quotas for each quota name referenced in the specifications: "name", target "name.T", and resettable "name.R". "TRIPLE_QUOTAS= MOD_TARGETS_ONLY" means supervisors can only modify the target quotas on QSS screens; they can't modify the standard or resettable quotas.

Programming/Size-Related Options

Statement	Explanation
ANSWER_LENGTH=#	Sets “answer array” size (default is 10000 bytes; minimum is 1000; maximum is 60000. This is the allowable # of characters in backreferences to questions.
CASE_LENGTH=#	Sets available space for the respondent’s answers. Default is 800 columns.
FONE_TEXT_LENGTH=#	Specifies the number of columns of user text in the phone file. The default is 200, the maximum is 4900.
MAXIMUM_GRID_QUESTIONS=#	Allows you to have more than the default of 200 questions in a grid.
MAXIMUM_LABELS=# #	# of question labels allowed. The default is the maximum of 32000 labels, setting it smaller makes a smaller questionnaire file.
MAXIMUM_QFF_FILE_ SIZE=#	Specifies the maximum size in megabytes for the QFF file. The default is 1 megabyte and the maximum value is 50 megabytes.
-MONITOR	Specifies maximum question size. The default size is 32000 bytes, the maximum 64000.
MAXIMUM_QUOTA_NUMBER=#	# of numbered quotas, if used (100-2.1 million).
ROTATE_SEED_WIDTH=#	Specifies the allowable size of !Rotate seeds for !Rotate,S statements. The default is 10, but this feature lets you set it to a smaller value. The purpose of this is to allow older spec files to retain the old maximum of six characters. The maximum was increased to 10 to accommodate more questions per rotate. A maximum of 10 ensures that all possible combinations of larger rotates will be random. With a six-digit random seed, this could not be guaranteed.

Statement	Explanation
SCREEN_LINES=#	Maximum screen size for a question. Default is 23. WebSurvent users should set this to 200 or so.
SPEC_WIDTH=#	Defines the max line width in the questionnaire spec. The default is and maximum is 5000, and the minimum is 72. Set this to 80 if you want to check that all screens will fit on an 80-column screen.
TEXT_START=#	Stores the data from TEX questions beginning at the location #.
TRFILE_DIRECTORY=#	Controls the max number of case ids in the datafile. The default is 20,000.
WORK_START=#	Specifies a data area where the system will not change the next data column pointer after questions with locations are specified.

Programming/Compile Related Options

Statement	Explanation
ALLOW_(DATA)_OVERLAP	This command allows you to have questions that start in the data area of a prior question and go beyond the end of the data area for that question, or that start before a question and overlap into another question's data area. The default is that this is not allowed, because you are probably mistakenly overwriting data.
-CHECK_FILE	Suppresses the creation of a list of the compiled questions and their data locations (the CHK file).
DATA_LOCATION_REQUIRED	Requires all data-asking questions to have a data location specified.
DATA_OUTSIDE_LOOP_OK	Necessary if you want to have some LOOP variables save data outside the LOOP area.
-DB_FILE	Suppresses the creation of a variable data base (DB) file.

Statement	Explanation
-HARD_COPY_FILE	Suppresses the creation of a hardcopy (HRD) file (readable listing of questions and logic in the questionnaire).
MAKE_NAMES	Tells the program whether to make names for unnamed variables.
NUMERIC_WIDTH_REQUIRED	Requires a width on all data location specifications for all EXPR and NUM questions without a range.
QFF_FILE_NAME=name	Specifies a QFF filename studyname.QFF, so you can run two questionnaires accessing the same data, phone, and quota files.
-QUOTA_FILE	Suppresses the creation of a quota file (QUO).
QSP_FILE	Suppresses the creation of the backup specification file (QSP).
SPL_DATA_LOCATIONS_OK	Allows old-style HP3000-SPL specification type format for data locations e.g., A01 for record one, column one, or C23 for record three, column twenty-three.
-SUMMARY_FILE	Suppresses the creation of a summary list which contains a one-line description of each question (the SUM file).
VALID_WIDTH_REQUIRED	Forces the questionnaire writer to match the proper width the program would assign on all questions.

QUESTION STRUCTURE: (2.3.4 Question Structure)

```
{label: location options       (only "{" required)
!RDG                           random data responses
                               weights (optional)
!MISC                          MENTOR spec output control
                               (optional)
!HELP                          Personalized help message
                               (optional)
!IF condition                  (optional)
     question text             (optional)
     !question type, subtype, options
     response list             (if required by type)
```

```
        }
```

On the question label line, label can be up to 8 characters, "location" can be ###.##, [previous label] to put in prior label's location, or WORK to put in WORK area. It can include an "offset", eg. [label+2.2]. "Options" can be "HIDE" to hide the question or "ALIAS=label" to treat this like the answer to a prior question.

DATA ENTRY QUESTION TYPES: (2.4 DATA ENTRY QUESTION TYPES)

Category (CAT). Uses a list of pre-defined responses which may allow single or multiple responses. Each response places a binary punch (1, 2, 3, 4, 5, 6, 7, 8, 9, 0, X or Y) in the data. There may be up to 240 response codes. The maximum data width is 240. You can rotate and control list by prior question responses.

Examples: Rating scales, products used.

Field (FLD). Same as CAT, FLD uses a list of pre-defined responses, but stores the response code(s) in the data. Multiple responses are allowed. The maximum data width is 3000, max codes approximately 800.

Examples: State codes (CA,NY,TX), ZIP codes.

Numeric (NUM). Used when the response can be a number within a range. The numeric value is placed in the data. You can specify up to three exception codes. The maximum data width is 16. *Examples:* Household income, age, price.

Text (TEX). For long or extremely variable-length, open-ended responses. Up to 5000 (UNIX, 2000 DOS) characters may be entered. The response is placed at the end of the data file in compressed format. A full-screen editor (monitors) or line editor (terminals) is used for data input. *Examples:* General likes/dislikes, other uses for product.

Variable (VAR). For short, controlled-length, open-ended responses. The response can be from 1 to 76 characters long. Minimum and maximum number of characters are controlled.

Examples: Name, address, zip code.

QUESTION TYPE LINE: (2.4 DATA ENTRY QUESTION TYPES)

Syntax:

```
    !type, subtype, options
```

Data entry types: Category, Field, Numeric, Variable, Text

Control types: Display, Edit, Expression, Generate, Goto, Loop, Phone, Quota, Reset, Special, System, Zspc

DATA ENTRY QUESTION LINE, SUBTYPES: (2.4.1, 2.4.3, 2.4.5, 2.4.6 and 2.4.7)

!CAT,subtypes,number of responses allowed, rotate_seed=xxx or

IR=/IL=/XL=/XR=/;eie,rotate_seed=xxx

Subtypes:

A Create a hidden response list filled by a data generation or a previous question

B Allow blank

C Display only responses that have *not* had data generated to them from a prior question

D Display only responses that *have* had data generated to them from a prior question. Clears the prior data after new response is entered

I Include or exclude previous answers. Extra parameters are: IL=<labels to include>, IR=<responses to include>, XL=<labels to exclude>, XR=<responses to include>

M On Highlightcats list, use response text instead of code to control movement

N Hide the response list during the interview

R Rotate the response list before displaying it. Use the "ROTATE_SEED=xxx" parameter to have 2 questions rotate in the same order

S Sort response list by response code

T Sort response list by response text

X Re-enter responses and not overwrite existing codes

Combinations of subtypes are specified by entering the codes in a row.

!FLD,subtype, number of responses allowed, rotate_seed=xxx or

IR=/IL=/XL=/XR=/;eie,rotate_seed=xxx; or <dbrfilename>,<prefilled response>

Subtypes:

A Create a hidden response list filled by a data generation or a previous question

B Allow blank

I Include or exclude previous answers. Extra parameters are: IL=<labels to include>, IR=<responses to include>, XL=<labels to exclude>, XR=<responses to include>

M On Highlightcats list, use response text instead of code to control movement

N Hide the response list during the interview

R Rotate the response list before displaying it. Use the “ROTATE_SEED=xxx” parameter to have two questions rotate in the same order

S Sort response list by response code

T Sort response list by response text

X Re-enter responses and not overwrite existing codes

If a disk-based response list is referenced, \|label| lets you prefill the response with the value in label. Combinations of subtypes are specified by entering the codes in a row.

!NUM,subtype,NUMDECS=#,range,other number,exception codes

Subtypes:

B Allow blank

R Displays rating scale centered on zero to +/- the maximum value (|-----*-----|)

V Allow varying numbers of decimal significance

X Save space for a plus or minus sign for database export

Z Zero-fill the response in the data

NUMDECS: 1-8 *range:* <label, location, or number> - <label, location, or number> *other number:* single number

Exception codes: Up to 3 one to 16 alphanumeric character codes, or *X or *Y for punch X or Y

!VAR,subtype,maximum,minimum

Subtypes:

A Create a hidden response filled by a data generation or a previous question.

B Allow blank

E Checks for a valid email address

L Allow alphabetic characters and blanks only

N Numeric responses only

P Phone number format only, (1-(xxx)yyy-zzzz or such)

Q Doesn’t show response on screen; used for passwords

Minimum: # of characters, 0-76 for input; default is 0

Maximum: # of characters, 1-76; default is 1

Hidden !var maximum is 3000 columns

!TEX,subtype,# of lines, # of columns (full-screen mode)

or

!TEX,subtype,# of characters (line mode)

Subtypes:

A Create a hidden response filled by a previous question

B Allow **ESC** (DOS; **Enter** for UNIX) as a blank response

L Force line edit mode like UNIX

# of lines: 1 to the number of lines on the screen minus two. Default is 9.

# of columns: 10-78 (terminal mode). Default is 78.

# characters 1-5000

QUESTION TEXT: (2.5 QUESTION TEXT)

<u>Screen controls: *(2.5.1 Screen Format Controls)*</u>

_ Place the cursor for data entry here (no prompt displayed unless specified before the _)

\N Skip to a new line

\(#,#,#,#) Screen position for question; numbers are: top row, left column, bottom row, right column

\((#,#,#,#) Screen position for this and subsequent questions

\A Use the next available line on the screen

\(BR) Position what follows on bottom of screen

\(BR-#) Position what follows on bottom -# lines of screen

\H Wrap the response list horizontally

\T Clear the screen

<u>Centering and outlining options</u>

\O# Outline whole question and list with box (#=1-3)

\OR# Outline recode table with box (#=1-3)

\OCT Center question text horizontally

\OCR Center response list

\OD# Default outline question and response list (#=1-3)

\ODCT Default center text (not response list)

\ODCR Default center response list

\ODT# Default outline text of question

\ODR# Default outline around response list

\OD0 Turn off all default outlines and centering

__**Mono Text Enhancements:** ***(2.5.2)***__

\B Bold

\E End enhancement

\F Flashing

\I Inverse video

\U Underscore

__**Color Text Enhancements:** ***(2.5.2)***__

\Cfb Color foreground/background

\DCfbCfb Color for entire question; first Cfb set is for text; second Cfb set is for the response list

\DDCfbCfb Default color for this and subsequent questions

\E End enhancement

__**Color set:**__

B blue

C cyan

G green

M magenta

R red

W white

Y brown

Z black

__**Displaying Special Characters:** ***(2.5.2)***__

\^## Display graphics character with hexadecimal value ##. (See *Appendix E*.)

\^[Display escape sequence.

\^ Display control sequence.

Displaying Prior Responses and Data: *(2.5.3)*

\:QQ# or \:label Display single response to prior question

\:label^# Display response using # screen positions.

\:label^-# Display response using # screen positions, right-justified.

\#(label or QQ#, first response, number of responses) Display multiple responses.

\|location| Display contents of data location. Location can be absolute location, record/column.width, or :label:.

\[col.wid] Display data from the phone file or suspend text (location 20001.80)

Miscellaneous Displays: *(2.5.4)*

\@ Display question number.

\- Send following text to DB file only, not to Survent screen.

\+ Send following text to Survent screen only, not to DB file.

***** Send following text to both Survent screen and DB file.

Multiple Languages: *(2.5.5)*

\Lxx Switch to chosen language to display

CONDITION STATEMENT FUNCTIONS: (2.6.4 Using Functions in Condition Statements)

FUNCTION NAME**(**arguments**)**

<u>Data Generating Functions</u>

IFTHEN, IFTHENELSE

Returns numeric values and expressions based on some condition

```
EX:
!EXPR,,,IFTHEN(QQ5>10,50)
```

LOCALSCRATCH(start, length)

Returns the contents or a region of the local scratch area

```
EX:
!IF LOCALSCR(1,2)="34"
```

MAX(numeric expression 1,numeric expression 2, numeric expression n)

Returns the highest number from the numbers specified.

```
EX:
!IF MAX(23,[10.3],AGE) > MIN(QQ1,QQ2)
```

MIN(numeric expression 1,numeric expression 2, numeric expression n)

Returns the lowest number from the numbers specified. See MAX above for example.

RANDOM(#)

Returns a random number between zero and the number specified.

```
EX:
!IF Random(9)+1 > 3
```

X(label or location)

Returns zero if the question’s response is blank or non-numeric, otherwise the number

```
EX:
!IF X([10.2]) + X([AGE]) > 33
```

Response Checking and Counting Functions

CHECKTEXT(TEX question label or location)

Returns the number of characters entered as a response to a TEX type question

```
EX:
!IF CHECKTEXT(OthCars) > 0
```

NOT(condition**)**

Returns true or false, opposite of the result of the condition.

```
EX:
!IF NOT((AGE > 34) AND SEX(M))
```

NUMITEMS(label or **[**location**])**

Returns the number of “punches” in a label or location.

```
EX:
!IF NUMITEMS(CARTYPES) > 1
```

XF(MATCH_TEXT(TEX or VAR label or location, string 1 , string 2, string n**))**

Returns the position of first string found in the answer to VAR or TEXT question

```
EX:
!IF XF(MATCHTEXT(Q23,"COLA") > 1
```

XF(NUMBER_OF_RESPONSES(label**))**

Returns the number of responses in a CAT or FLD variable.

```
EX:
!IF XF(NUMRESP(BEV1)) > 1
```

XF(TIMES_ASKED)

Adds 1 to a number every time you go over a question with this function. If you go over it once, back up, go over it again, it returns a "2".

```
EX:
{Times: .2 !expr,,,Xf(Times_asked)}
```

XF(STRING_TO_NUMBER ()) turns strings into numbers. This function reads any variable or function and returns it as a numeric value if the result is a number. For instance, it will read "1" and return it as " 1" (a right-justified numeric 1). *Refer to Chapter 6* for more on this function.

Quota-Related Functions

MODQUOTA(quotaname)

Returns the change in the named quota since the beginning of the interview

```
EX:
!IF MODQUOTA(MALES) > 1
```

MODQUOTN

Returns the change in the numbered quota

```
EX:
!IF MODQUOTN(456) > 1
```

QUOTA(quotaname)

Returns the value at the beginning of the interview of the named quota

```
EX:
!IF QUOTA(MALES) >= QUOTA(MALES.T)
```

XF(QUOTA(<[label or location**]** or <quota label>**))**

Returns the quota value of a quota whose name has been specified in a prior question

```
EX:
{!if xf(quota([Quoname])) >= xf(quota([Targname]))
```

QUOTN(quotanumber, location or label)

Returns the current value of the numbered quota specified.

```
EX:
```

```
!IF QUOTN([123]) > 82
```

QUOTANOW(quotaname)

Returns the current value of the named quota

```
EX:
!IF QUOTANOW(NEWRESP)<=QUOTA(NEWRESP.T)
```

<u>**Phone File Related Functions**</u>

FONESTATUS( )

Returns the current status of the phone record

```
EX:
!IF FONESTATUS()=1
```

FONETEXT(column, width)

Returns the contents of the specified field in the phone file

```
EX:
!IF FONETEXT(60,3)="ZOO"
```

LAST_CALL()

Says whether you are on the "LAST CALL" for the phone number you are calling

```
EX:
If LAST_CALL()
```

XF(MARKET_WEIGHT(<marketname or **[<**label or location**>]>))**

Returns the value of a particular market weight

```
EX:
!IF XF(MARKETWEIGHT([Wghtval])) = 0
```

XF(MAX_ATTEMPTS)

Returns the "Maximum Attempts" setting from the phone file

```
EX:
{maxatt: .3 !EXPR,,,XF(MAX_ATTEMPTS)}
```

XF(NUMBER_REDIAL)

This returns how many times the number has been re-dialed during this interviewing session either using the "REDIAL" command or automatically.

Date-Time Functions

XF(DATE_TIME_DIFFERENCE(date1, date2, type returned **))**

Returns the difference in 2 date/time strings in yrs, mnths, wks, days, hrs or minutes

```
EX:
XF(DATE_TIME_DIFF( born, today, DAYS))
```

XF(DATE_TIME_OFFSET(location, date to compare, type of return, # to offset by **))**

Return a date in the format YYYYMMDDHHMMSS given a starting date and offset

```
EX:
XF(DATE_TIME_OFFSET(newdate,olddate,weeks,15))
```

XF(DAY_OF_WEEK(date **))**

Returns the day of the week associated with the date specified

```
EX:
{DOW: .1 !EXPR,,,XF(DAY_OF_WEEK(DATE)}
```

XF(TIMED_CALL) returns how many numbers are scheduled for a certain time so far. This function can be used if you are trying to schedule calls and want the interviewer to be able to know how many calls have been scheduled at that time. Typically, this would be used in an !expression statement.

TIMEDIFF ([loc1.12$] or label$,[loc2.12$] or label$)

Returns the difference of two times in seconds

EX: !IF TIMEDIFF([10.12$],[30.12$]) > 60

Mode-Checking Functions

DATAGEN( )

Returns a “1” or “true” if the interview is using the random data generation operation

```
EX:
!IF DATAGEN()
```

DIALER( )

Returns “1” if using a predictive dialer, “2” if power dialer and “0” (“false”) if not

```
EX: !IF DIALER()
```

XF(DIALER_TYPE( ))

Returns dialer type (0=no dialer, 1=SER, 2=PRO-T-S, 3=NOBLE, 4= Stratasoft)

```
EX:
{Dialt: .1 !EXPR,,,XF(DIALER_TYPE)}
```

System-Related Functions

XF(USABLE_STUDY(questionnaire filename **))**

Returns a “1” if the questionnaire file is present, and a “0” if not. This is usually used when you have a list of studies for the interviewer to choose from and have !CALL statements to execute the questionnaire if it exists.

XF(PROCESS_ID()) (Unix)

Returns the process ID of the current survent session. This can be used for accounting or

to remove process IDs that are not active

CONTROL STATEMENTS: (3.1 CONTROL STATEMENTS)

Display (DISP) statements display text on the screen, with or without a pause for the interviewer to read it.

!DISPLAY,subtype,#

Subtypes:

1 Display text and wait for **Enter** to continue. This is the default.

2 Display text, don't wait for response.

3 Display text, pause for # seconds.

6 Like subtype 3, but allows interviewer to press <Enter> at any time.

#: Number of seconds to pause if subtype 3.

Edit (EDIT) statements edit prior TEX and VAR questions.

!EDIT, <u>TEX or VAR question label to edit</u>

Expression (EXP) statements do calculations using arithmetic and special functions.

!EXPR, subtype, numdecs, <u>expression</u>

Subtypes:

B Store blank-filled numbers (default) (e.g., " 12")

S Puts the specified "string" in the data

Z Store zero-filled numbers (e.g., "00012")

Numdecs: can be from 1 to 8 or NODECIMALS. If you do not specify the number of decimals, the default will be the highest number of decimals on the expression.

Expression: A numeric expression using data locations, labels or numbers combined with math joiners or functions (see !IF condition), especially + - / * /and **; or a string to be stored in the data.

Generate (GEN) statements add, remove, or net responses to CAT questions, or move or blank data.

!GEN, subtype, label or location, question references

Subtypes:

B Blanks a question

M Moves (copies) a question

U Removes TEX pointers and their associated data

A Adds a code to a response list

Z Zaps (removes) a code from a response list

O Nets codes from one question to another

T Takes codes out of one question that are in another

Q Replace code in CAT/FLD with codes from other questions

R Replace code in CAT/FLD with other codes

E Erases the answer array

C Compresses data

Question references:

A or Z: label or location, response code or punch to add/zap

```
EX:
!GEN,A,BRANDS,3
```

B: label or location to blank, length

```
EX:
!GEN,B,[2/1],80
```

C: label or location

```
EX:
!GEN,C,[1/50.10]
```

E: label

```
EX:
!GEN,E,REMEMBER
```

M: TO label or location, FROM label or location, length to move, offset

```
EX:
!GEN,M,NEWBRAND,BRAND
```

O: TO label or location, FROM label or location, -length to net

```
EX:
!GEN,O,NEWBRAND,BRAND
```

A '-' before the length nets all the columns of the "from" question into the first column of the "to" question.

T: TO label or location, FROM label or location, length to take out

```
EX:
!GEN,T,NEWBRAND,BRAND
```

Q: TO label, Code to replace, Questions replaced FROM

```
EX:
!GEN,Q,NEWBRAND,99,BRAND1,BRAND2
```

R: TO label, Code to replace, Codes to replace with

```
EX:
!GEN,R,NEWBRAND,99,23,24
```

U: label or location

```
EX:
!GEN,U,OTHRBRND
```

A GRID question block is a set of questions all of which are presented on the screen at the same time.

!GRID, subtype

Subtypes:

B Allows blank responses in the grid

R Re-executes all the grid questions after each response

E Lets you edit a previous grid

M Says you must answer all questions before you can get out (default)

!END_GRID can have an IF condition attached to it which must be satisfied before you can get out.

Goto (GOTO) statements are used to skip to a question based on a given condition, or are moved to.

!GOTO,label

Loop/Endloop (LOOP/END_LOOP) statements are used to repeat a series of questions for each response to a multiple-response CAT (or FLD) or a numeric (NUM or EXP) question, or each line of question text for a GOTO question.

!LOOP,<u>control question label</u>, maximum times through loop, amount of data space per loop

!END_LOOP

Quota (QUO) statements add to a counter for quota requirements. These are most useful in a networked system.

!QUOTA,<u>subtype</u>, Quota to increment, # to increment, NOW

subtypes:

1 Update standard quotas. This is the default.

2 Update numbered quotas.

quota to increment: name or [location] in data of name to update ([location or label]) if subtype 1, or number/[location] of number to update if subtype 2.

The *number* may be a number, or a label or location where the number to update with is stored.

NOW: Increment the quota immediately, instead of at the end of the interview.

Reset (RSET) statements return the interviewer to a previously asked question and clear intermediate responses.

!RESET, <u>label or -#</u>

If you use - followed by a number, the interview will be sent back that many executed questions.

Special Information (SPC) statements get information from the system or perform other special functions.

!SPC, <u>subtype</u>, options

Subtypes:

3 Stores current date and time in the data in the format YYYYMMDDHHMMSSdJJJ in version 7.6L and 7.7.

4 Stores current date and time in the data in the format DDD MMM DD YYYY HH:MM xx.

5 Stores employee file information in the data.

6 Stores the time in seconds from the start of the interview.

7 Stores the interviewer ID, interview type (A/S/B), date/time MMDDYYHHMM site code/dialer #, station number, study name, special interviewer types, interviewer time zone, terminal type, phone extension, qfile time stamp, consecutive backups, TTYINFO file info, four-digit year date/time, and language code. Here are the first 64 characters:

```
0        1         2         3         4         5         6
124567890123456789012345678901234567890123456789012345678901234
FRED 0307941320 SF 00002 COMP2   38        08       0306941013
```

***9** Causes the text line on this question to be saved as a response. The option is the controlling label or location with the number of the text line to get (not required).*

***A** Assigns the case ID now, rather than at the end of the interview. Can also assign a “unique number” with subtype “A,N”.*

***B** Causes the interview to terminate, without saving the data.*

***D** Sets the case ID to the data found in the columns specified for the case ID in the header statement.*

***E** Displays text on the SURVSUPR screen or sends it to the SERVER. Subtypes 1,2 and 3 write the info to the DAI screen, the supervisor, or log file respectively.*

***F** Puts the station number into the data (up to 4 characters).*

***H** Special abort and write case options*

***0** Default. Abort this case and don't require an Enter to start the next one.*

1 Write this case to disk with a case ID of NOTKNOWN (or as many of these letters as will fit in your case ID field). Don't update any quota cells, and don't update completes in SURVSUPR's status screen. Any quota changed by a QUOTA NOW will still change the cell value.

2 Do not allow SUSPEND from the time this question is executed until the end of this interview or until an SPC,H,3.

3 Undo an SPC,H,2; allow SUSPENDs again.

4 Disable ^ from here on (until SPC,H,5).

5 Re-enable ^ after SPC,H,4.

6 Disable RESET from here on (until SPC,H,7).

7 Re-enable RESET after SPC,H,6.

8 Disable RETAKE from here on (until SPC,H,9).

9 Re-enable RETAKE after SPC,H,8.

A A, seconds: sets the FORCE_INTO_INTERVIEW time to "seconds."

J Specifies a pause of n seconds.

K Puts information in the local scratch area, which stays there across interviews for a particular interviewer.

L Gets information from the local scratch area, which had been stored there with an SPC, K statement from this or a prior interview.

M Puts information in the global scratch area, a place in the quota file used to hold things other than quota numbers. This holds across all interviewers.

N Gets information from the global scratch area, where it had been stored with an SPC, M statement in this or in a prior interview.

P Gets a case for Survent to use, typically for cleaning or coding. Suboptions: 2-5, 2=return to interview, 3,4=retrieve new case ok,5=get case checked out already ok

R Sends the data specified to a record in the specified “savelog” file

S Moves data to the suspend text.

T Executes the Special block of questions immediately.

U Does a programmatic SUSPEND on the next question.

V Stores the question number or label of the previously executed question.

W Immediately writes a case. Subtype “N” is the default and writes a case id of “NOTKNOWNID” to the file unless !SPC,A or !SPC,D has executed. Subtype “U” updates the current case with new info. Subtype “X” creates a new caseid for each record written.

Y Causes Survent to provide an answer for the next question it sees if an answer is not supplied in a certain number of seconds.

Z Accumulates interview time.

0 Start the timer

1 Stop the timer

2 Read stop watch (record time) but don’t turn it off

3 Get time since last SPC, Z, 0, or 1 without resetting the time.

SYSTEM STATEMENTS

(SYS) statements get information from the operating system (DOS/UNIX) or set system level values.

!SYS, <u>subtype</u>, values

1 Send lines of text as program commands.

2 Send lines of text as operating system commands

3 Send commands to a com port on a PC

L Set the value for the language to use, where language is “xx”

Q Causes the Survent session to terminate at end of interview; this will run the AFTER_QUIT block if one is programmed.

<u>CALL STATEMENTS (3.1.6 Call Questionnaires)</u>

Call statements (CALL) call other questionnaire to be run as part of the current interview.

!CALL, subtype, questionnaire file to call

1 Call a questionnaire and save a separate data case, but come back to the controlling interview when you are done. The called questionnaire will use the main questionnaire's quota and phone files.

2 Call a questionnaire, but give total control to the called questionnaire, using its own quota and phone files.

3 Like subtype 1, except it doesn't load the CALLed questionnaire under the server until it is actually requested.

4 Like subtype 2, except it doesn't load the CALLed questionnaire under the server until it is actually requested.

DISK-BASED FIELD QUESTIONS: (3.1.4 Data Entry Question Function Modifying Statements)

Maximum 20,000 categories, only one result code allowed, for large brand lists.

Syntax: **!FLD**, subtype, ,filename, pre-fill reference

Subtype: A or none

Filename: Any valid file name, it will use a .dbr extension unless *$filename* is used.

Pre-fill reference: A backreference to codes from prior questions or locations using \|label| syntax

Syntax for disk-based file:

```
~PREPARE DISK_BASED_RESPONSE_LIST (or DBR)
OUTPUT=filename
      !level1 text
Rcode response1 rtext
      !level2 text
Rcode response2 text
END
~END
```

!level text: not required, displays for that level at screen

Rcode: Result code, what gets stored in the data, 1 to 19 characters.

Response1: Response code, a '-' after response code doesn't save text in answer array.

Rtext: Text for the response.

END: End of DBR specifications.

COMPILER COMMANDS (3.2 QUESTIONNAIRE CONTROL USING COMPILER COMMANDS)

{!AFTER_QUIT}

Starts a block of questions that will be executed at the end of the interviewer's session (when they type "Quit" of if they execute a !SYS,Q command. It will save a separate datacase. Use {!END_AFTER_QUIT} to end the block.

{!ALLOW_ABORT}

Allows the interviewer to terminate the interview by entering ABORT. The default is not to allow ABORT. The command "-ALLOW_ABORT" will disable ABORT until the end of the interview or ALLOW_ABORT is re-specified.

{!ALLOW_BACKUP}

Re-enables backing up in the interview using the caret (^) after this point. This is the default. -ALLOW_BACKUP will disable ^ until the end of the interview or an ALLOW_BACKUP is specified.

{!ALLOW_MONITOR}

Allows monitoring of the following part of the questionnaire by SURVSUPR or SURVMON. This is the default and need only be specified after a -ALLOW_MONITOR has disabled monitoring.

{!ALLOW_RESET}

Re-enables the Survent interview command RESET. This is the default. -ALLOW_RESET will disable RESET until the end of the interview or an ALLOW_RESET is specified.

{!ALLOW_RETAKE}

Re-enables the Survent interview command RETAKE. The default is not to allow RETAKE. -ALLOW_RETAKE will disable RETAKE until the end of the interview or an ALLOW_RETAKE is specified.

{!ALLOW_SPECIAL}

Re-enables the SPECIAL command to go to a SPECIAL block and return to the question after. –ALLOW_SPECIAL disables the SPECIAL command until the end of the interview or an ALLOW_SPECIAL command is specified.

{!ALLOW_SUSPEND}

Re-enables the Survent interview command SUSPEND. This is the default. -ALLOW_SUSPEND will disable SUSPEND until the end of the interview or an ALLOW_SUSPEND is specified.

{!ALLOW_TERMINATE}

Re-enables the Survent interview command TERMINATE. This is the default. -ALLOW_TERMINATE will disable TERMINATE until the end of the interview or an ALLOW_TERMINATE is specified.

{!AUTO_PUNCHES}

Makes category questions get written out in the QSP file as CAT* questions. The response codes determine the punches to be stored, e.g., a response code of 10 would cause a 0 punch (tenth punch of the field) to be stored. Use -AUTO_PUNCHES to turn this off.

{!AUTO_RESPONSE_CODE=#}

Generates response codes for items in response lists (they will be specifically shown in the QSP file). The codes will start at 1. #=1 will return codes 1-9, 0, A-Z to a maximum of 36 codes, =2 will return 01-99, =3 will return 001-250.

{!AUTO_RETURN}

Causes responses to be accepted as soon as they reach their maximum length. The standard prompt changes to ==> in this mode. Use -AUTO_RETURN to turn off this command.

{!BACKUP_HERE}

Will back up to this question from any following question.

{!BACKUP_WHEREAT}

Will back up to the previous question from now on (default).

{!BLANK_LINES=#}

Puts # blank lines after every question’s text.

{!BLOCK #}

Saves # columns for the next block of questions. This must be followed by question specifications and then END_BLOCK.

{!CHECK_COLUMN_OVERLAP}

Turns on PREPARE’s checking of data column overlap. This is the default. -CHECK_COLUMN_OVERLAP would turn the checking off.

{!COLUMN_KICK #}

server_Causes # blank columns before the question’s response — useful for reserving columns for future questions.

{!COLUMN #}

Causes the next question’s answer to be assigned to column #.

{!COMMENT}

Leaves information after the !COMMENT but before the } as a program comment.

{!DEFAULT_NUMERIC_MINIMUM=#}

Allows you to specify what value to default to as the minimum value in NUM questions; the default is “0” but you may set it to “1” or “MINIMUM_NEGATIVE”.

{!DO_MENTOR}

Turns on the making of variables in the DB file and the makingof MENTOR spec files. Use –DO_MENTOR to turn it off. If the header option -DB_FILE is specified, you will get no variables regardless of this setting.

{!DO_VARIABLES}

Turns on the making of variables in the DB file. This is the default. Use -DO_VARIABLES to turn it off. If the header option -DB_FILE is specified, you will get no variables regardless of this setting.

{!ECHO_CATS}

Places you in Echocats mode, where responses to CAT questions will be highlighted. The default is to not be in Echocats mode. -ECHO_CATS will disable Echocats mode until the end of the interview or an ECHO_CATS is specified.

{!END_AFTER_QUIT}

Ends the After_quit block. Required with AFTER_QUIT.

{!END_BLOCK}

Ends the block that began with the last BLOCK. Required with BLOCK.

{!END_REMOVE}

Ends the Remove block. Required with REMOVE.

{!END_RESUME}

Ends the Resume block. Required with RESUME.

{!END_ROTATE}

Ends the Rotate block. Required with ROTATE.

{!END_SPECIAL}

Ends the Special block. Required with SPECIAL.

{!END_SUSPEND}

Ends the Suspend block. Required with SUSPEND.

{!ERROR_LEADIN text}

Allows you to replace the text “ERROR #” on error messages.

{!ERROR_MSG #### text}

Allows you to replace message number #### from the CfMC MSGFILE with your own text.

{!ERROR_STOP #}

Stops subsequent compiles (using ~PREPARE COMPILE) after # errors.

{!EXPORT_LEVEL=#}

If the export level is <= the level specified in the reformat run for a particular question, the data from that question will be included.

{!FILL_VIEW}

Forces responses from the data instead of being generated from the interview when in VIEW mode.

{!FIX,#}

Within a Rotate block, fixes the position of the next # questions. If used, this must follow a ROTATE command. If # is omitted, all questions until the next relevant command are fixed.

{!FLAG_DISALLOWED_CATS}

Turns on the flagging of categories that are not allowed on CAT or FLD question lists (typically when using subtypes I, C, or D). Turn back off again by using -FLAG_DISALLOWED_CATS.

{!FLD_SAMEAS_USE_ORIGINAL_WIDTH}

Forces the programmer to use the same widths for questions as the program would otherwise generate.

{!GROUP,#}

Within a Rotate block, keeps the next # questions together as a group when they are rotated. If used, this must follow a ROTATE command. If # is omitted, all questions until the next relevant command are grouped.

{!HARD_CODE}

Hardcodes the data columns for the questions in the backup QSP file after a compile. Read the QSP file into a compile to permanently fix the locations. Use - HARD_CODE to turn off.

{!HARD_COPY option}

Produces a hardcopy listing of the questionnaire, including all questions after the HARD_COPY command. Creates the file <study name> with an extension of HRD. Use -HARD_COPY to turn off.

{!HELP_<question type>,<help message>}

Puts a HELP message at the bottom of the interviewer's screen for the question type(s) you specify. There are default messages.

{!HELP_ON_TOP}

Puts the line derived from the !HELP_type command or the !HELP line on a question at the top of the screen instead of at the bottom.

{!HIDE_ALL}

Hides all subsequent questions. Use -HIDE_ALL to turn off.

{!HIGHLIGHT_CATS}

Creates an interactive screen for CAT and FLD questions. The user is positioned on the first category, which is highlighted. To move to another category, the user types the response code or text, or uses the arrow keys to move to the relevant category. To accept the category as the response, the user presses **ESC** or **Enter**. Use -HIGHLIGHT_CATS to turn off Highlightcats mode.

{!HIGH_POINT x}

Saves the current location as x, if it is higher than a previous one saved as x. X is a letter from a-z.

{!INCLUDE filename}

Specifies the name of a file to be included in the compile.

{!KEEP_RESPONSE_CODES}

Cause back-references to questions to include the response code instead of just the response text.

{!LANGUAGE}

Specifies the languages used and character set to allow in the questionnaire specifications. Languages are defined by “SET=(lang1 lang2), character set options are:

CHARSET=ASCII, EXTENDED_ASCII, MULTIBYTE (Chinese/Korean), and SHIFT_JIS (Japanese).

{!MAKE_SPEC_FILES}

Turns on the making of specification files (Mentor, SPSS, etc.). This is the default. Use -MAKE_SPEC_FILES to turn off. This does not affect the making of variables for the utilities (see DO_VARIABLES). This command has no effect on the QSP, CLN, SUM, or CHK files. You must be making the spec files for this command to have any effect.

{!MAXIMUM_CONSECUTIVE_BACKUPS=#}

Says how many backups in a row an interviewer can do.

{!MENTOR_CLN}, {!MENTOR_DEF}, {!MENTOR_TAB}

These commands control the information that gets passed to the CLN,DEF and TAB files.

{!NUMBERING #,#}

Controls incrementing for program-assigned question numbers.The first # is the starting question number which will be used for the next question. The second # is a number that will be the increment between question numbers. This command ontrols all subsequent questions until another NUMBERING command appears. The default increment is .10.

{!READ_NAMED_QUOTAS}

This does an immediate read of the quota file as it now stands. This eliminates the need for many QUOTANOW functions. Any subsequent QUOTA functions will reflect the values at the timeof the READ.

{!REMOVE}

Prevents compilation of a block of questions. REMOVE requires an END_REMOVE command to indicate where the block ends.

{!RESET_COLUMN #}

Removes references to existing question locations after the column specified (#). Use this command to reuse a data area over and over without worrying about specifying a location on each question.

{!RESPONSE_LIST_COLUMNS=#}

Enter a number from 1-4 to specify how many columns you want Survent to divide the screen into for questions with response lists.

{!RESPONSE_ON_RIGHT=x}

Reads specifications with the response code on the right of the text (i.e., Apples 01)

{!RESTORE_COLUMN x}

Resets the current location to the column saved as x with the SAVE_COLUMN x command. Subsequent questions are assigned columns beginning with the restored location.

{!RESUME} Causes the following questions to be asked only when an interview is resumed; these questions must be followed by an END_RESUME.

{!RESUME_HERE}

Interviews suspended after this statement return here when they are resumed.

{!RESUME_WHERE_AT}

Used after a RESUME_HERE to say resume at question suspended at from now on. This is the default.

{!RFT_CAT_01}

In conjunction with the REFORMAT utility, puts out responses to multiple-response category questions as 0’s and 1’s in successive columns, one column for each possible response; this is the default.

{!RFT_CAT_PUNCH}

Puts out single or multiple-response CAT question data as punches 1-9, 0, X, and Y in the RFT file.

{!RFT_CAT_RESPONSE}

Using the REFORMAT utility, puts out response codes for multiple-response category questions, up to the number of actual responses.

{!RFT_CAT_SPREAD}

Spreads multiple-response CAT data as punches, sequentially,in the recode table response order as one punch per column in the spread data file. This is like the default of '1s' and '0s'except that each column will either have the original punch or be blank.

{!RFT_ON}

Turns REFORMAT option on for succeeding questions (default); the following questions will be included in the RFT file when REFORMAT is run. Use -RFT_ON to turn off.

{!RFT_SAVE_LOOPS}

When using the REFORMAT utility, puts out loop block data in the RFT file sequentially starting at the first response found for as long as responses are found.

{!RFT_UNWIND_CODES}

Using the REFORMAT utility puts out CAT and FIELD data as consecutive codes. If a respondent answered "030501," the data would be made to look like "01bb03bb05" (where bbs are blanks).

{!RFT_UNWIND_LOOPS}

Using the REFORMAT utility puts out loop block data in the loops in its relative position by response code in the RFT file (i.e., the data in the third response would go in the third response position, even if it was the only response). This is the default.

{!ROTATE, option}

Begins a Rotate block, causing the questions in the block to be asked in some type of rotating order (scramble, random start, flip); the questions must be followed by an END_ROTATE.

{!ROTATE_ORDER label}

Stores the order of the questions asked in a following !ROTATE block to question "label". The syntax is:

```
    {!Rotate_order <label>[,asked,noblow] }
```

The question label to store the result in needs to be long enough to store the order, for instance, if you have 101 questions, it needs to be 3 * 101 = 303 columns long to store the order for all questions. "Asked" stores only the asked questions, "noblow" allows you to record only the first "n" responses without blowing due to lack of space.

{!SAVE_COLUMN x}

Saves the current location as x (a letter, so you may save up to 26 locations). You can then return to that location x later in the questionnaire with a RESTORE_COLUMN.

{!SCROLL_SCREEN}

Causes questions to display below prior questions on the screen by default instead of first clearing the screen. Use –SCROLL_SCREEN to turn this off.

{!SET_INTERVIEWER_CAPABILITIES=abcd}

Sets interviewer capabilities using the letters that correspond to the employee.xxx file columns 42-45. B=Break, C=Change interviews, D=Debug, E=Always echocats mode, L=logging, N=No capabilities, O=Log after every question, T=Training mode

{!SET_QUOTA name=#,OVERRIDE}

Sets the initial quota value in new quota files.

{!SHOW_LAST_RESPONSE}

Shows the response to the previous question on the current screen.

{!SHOW_QUESTION_LABELS=option}

Displays question labels in the lower right corner of the interviewer's screen. This is the default. Turn off by using -SHOW_QUESTION_LABELS.

{!SPECIAL}

This compiler command marks a block of questions for execution when an interviewer enters the keyword SPECIAL. At the end of the block (marked with {!END_SPECIAL} the interviewer will return to the point in the questionnaire where she/he had entered SPECIAL.

{!START_NEW_CARD}

Starts the next question on a new "card" when using CARD_ID formatted data with each 80-column data record having a caseid and cardid. See the CARD_ID header statement.

{!SUSPEND}

auses the following questions to be asked only when an interview is suspended; these questions must be followed by an END_SUSPEND.

{!TARGET name=#,OVERRIDE}

In conjunction with the TRIPLE_QUOTAS option on the header statement, this allows you to set the initial value of a target quota.

{!VIEW}

These commands will turn on or off the allowing of View.

{!VIEWX}

mode for a section of the questionnaire. !-VIEW says not to allow View mode until a !VIEW or !VIEWX is seen. !-VIEWX says not to allow View mode and not to execute any questions in this part of the questionnaire while in View mode.

{!VIEW_QUOTA=letter}

This controls how Survent will evaluate QUOTA functions read while in View mode. The default is that IF conditions with quotas on them will be false and EXP questions will return a '0' for the quota value. VIEW_QUOTA is overriden by -VIEW; if -VIEW is set, the whole section will not be executed. VIEWX after -VIEW will return control to the VIEW_QUOTA.

E Means 'evaluate', and it will return whatever the current

value is for the quota reference in the quota file.

F (false for IFs) and VIEW_QUOTA=N (O in EXPs) are the

defaults.

T will return True for IF conditions and EXP QUOTA( )references will return 1.

META COMMANDS (3.3.3 OPERATING SYSTEM COMMANDS and 3.3.4 PROGRAM CONTROL COMMANDS)

Meta commands always begin with the greater than symbol (>) and are directly followed by the command itself. Meta commands can be used in any CfMC program where it makes sense to use them. The following meta commands are particularly useful with Survent. For more information on these meta commands, their usage and allowable abbreviations, see the *Utilities* manual.

>BROWSE Displays a file on the screen.

>CFMC_EXTENSION Specifies whether to use standard CFMC file extensions or not. Can also say to use a different extension than the default for DEF, QSP etc. by adding DEF=xml for instance.

>CLOSE_DB Close an individual DB file.

>CREATE_DB Creates a DB file to store question entries. Options control the allowed size of the file. The DB file is kept open with ReadWrite access after it is created (see >USE_DB).

>DEFINE Defines variables as replacement for text or other references in spec files, eg. “>DEFINE @RANGE 1-5/6-10/11-20/21-99”.

>Define can also be used to do simple math to set the value of other defines. For instance:

Example:

```
>define @TEXTSTART 7001
>define @LASTDATA = TEXTSTART - 1
>echo @LASTDATA “will echo 7000”
```

Note that the “=” sign after the define. You can use other defines and any math statements in the defined variable (+. -, /, *).

>DUMP Sets switches to check internal processing information. Used for program debugging purposes or random data generation (see *2.7.2 RANDOM DATA GENERATION*).

>-EMIT_ALL_ASCII Converts , , etc. codes to "*" when Mentor writes ASCII files. You can now control the converting of special characters in the output file. >-EMIT_ALL_ASCII will cause all ASCII characters less than "30" on the ASCII code sequence get converted to "*". NUL, CR, LF and ctrl-z are all converted to '*' by default (with a message).

This is to allow CfMC programs to read records properly even if they have bad ASCII characters in them. For instance, is an "end of file" character that otherwise would cause the output file to end there.

>ERROR_AND_WARNING_SUMMARY Causes a summary of the errors and warnings in the run to print at the bottom of the listing.

>FORCE_HARDCODE Creates a hard-coded backup QSP file whenever a QSP file is created by PREPARE. Put this in the mentinit file if used.

>HALT Pauses the program; press **Enter** to continue. Options control where the pause occurs.

>IF_DEFINE Says whether to execute the following specifications based on whether a variable has been defined with >DEFINE.

>IF "@var"="string" Allows you to compare strings or numbers to control program flow. You may use "AND" and "OR" for multiple conditions. Use >ELSE and >ENDIF for multiple levels.

>LIST_DB_CONTENTS Lists all the entries in a DB file. If more than one DB file is open, you must specify which file to list.

>LOCATION_FORMAT Controls the format of locations printed in the CHK, HRD (hardcopy), QSP (backup), SUM files, the data location in Script Composer screens and Error and Warning messages.

>PRINT_REPEAT Controls whether the lines generated by a >REPEAT block will be displayed to the list file. PRINT_REPEAT will show the >REPEAT block expansion; this is the default. Use

>-PRINT_REPEAT to suppress the showing.

>PRINT_FILE Opens an ASCII print file to send just the output tables to for printing (with the extension of PRT, if none specified).

>PURGE_SAME Purges any existing file (except the file named at the List File prompt) that has the same name as any new file created by the program. Without PURGE_SAME, the program renames old files by changing the filename's first character to the next ASCII character. (AFILE, for example, would become BFILE.) Use >-PURGE_SAME to rename rather than purge; this is the default.

>QUIT Immediately returns to the operating system prompt from the program.

NOTE: With this command, PREPARE will not create backup QFF or QSP files, save datafile changes, or purge temporary files used by the programs.

>REPEAT/>END_REPEAT Sets a repeating pattern to produce multiples of specifications. You may specify nested repeats.

>RESET_DB Close all DB files.

>SAVEASFILE adds the process ID to saved file names. The process ID is added to the default "cc" file to make for example "cc". This is so multiple users can save their commands to different filenames even if you put ">SAVEASFILE" in the $CFMC/control/mentinit file. If the $USER variable is defined, we name the file "cc$USER". The purpose of this is to only keep one last copy of cc file around, not one for each process run (1/06).

>-SURVENT_DEBUG_INFO Remove the default information that displays to “debug mode” interviewers at the end of an interview.

>SYSTEM Operating system commands can be use inside CfMC programs by preceding them with >SYSTEM.

>USE_DB Opens a data base (DB) file. You can have several data base files open at the same time, but only one with ReadWrite access.

Options control access type and whether to write over old items with the same name.

>VER Lists the version number (and date) of the program which is currently being used.

PHONE STATEMENTS: (6.2.1)

PHONE (PHO) statements interact with the FON file by either getting a phone number to dial, displaying a status screen, or prompting for a number. (*6.2.1*)

```
!PHONE,subtype
```

Numeric subtypes:

0 Gets the phone number and displays call histories and status screen

1 Gets a phone number to be dialed by a preview mode dialer

M Sets the market weight

N Lets you get numbers that are not allowed by default when using PHONE,5 to get specific numbers

O Sets the special interviewer type in the current phone record

P Moves information from the data record to the phone record text area

R Use to send a phone number to a ROLM (or other) autodialer

S Specifies the status code for the current phone record when it is put back in the phone file

T Takes the time specified by the FROM location, converts it to respondent time, and puts it into the data location for this PHONE question

U Clears Ownership mode in the phone record and sets the current owner to blanks

X Marks the interview to be sent to a “validator” upon completion (See *Chapter 6: Independent Validation)*.

Z Moves data to the “call note” area of the phone history.

L Like PHONE,5, gets a number from a specific market for this interviewer

M Sets the market weight

N Lets you get numbers that are not allowed by default when using PHONE,5 to get specific numbers

O Sets the special interviewer type in the current phone record

P Moves information from the data record to the phone record text area

R Use to send a phone number to a ROLM (or other) autodialer

S Specifies the status code for the current phone record when it is put back in the phone file

T Takes the time specified by the FROM location, converts it to respondent time, and puts it into the data location for this PHONE question

U Clears Ownership mode in the phone record and sets the current owner to blanks

X Marks the interview to be sent to a “validator” upon completion (See *Chapter 6: Independent Validation)*.

Z Moves data to the “call note” area of the phone history.

APPENDIX Z: PRACTICAL APPLICATIONS

Z

COMBINING, USING SURVENT FEATURES

This appendix contains examples that show you how the different features of Survent work together in real situations.

How to Generate Data

The following scenarios show how to generate data for different CAT question subtypes.

GENERATING DATA FOR A CAT,A QUESTION

Often, you will collect data in different parts of a questionnaire that must be used to control further action in the questionnaire. You can accomplish this by generating responses into a CAT,A question based on condition statements, and using that question's responses to control a set of questions of possible places to go. Or you may just want to choose randomly from a list:

```
EX:
{Rand: .1
!EXP,,RANDOM(8)+1}
{
!GEN,M,WHICHCAR,RANDs}
{WHICHCAR:
!CAT,A,1
01 Chevy
02 Ford
.
.
09 Toyota}
{
Now we will be asking you questions about
\:Whichcar: ...
!DISP}
.
.
```

The EXP statement returns a random value from 1 to 9. Then the CAT,A associates some text with the number generated. The display question then displays this text based on the random number generated. The CAT,A is doing the association based on the punch in the data (1-9) and the punch assigned to WHICHCAR (since it is not specified, it defaults to a 1 punch for the first category, a 2 for the second, etc.), *not* the response code.

Or, you can use a CAT,A to modify the text on later screens based on previous responses.

```
 EX:
{NAME:
Enter the respondent's name
!VAR,,20,1}
{SEX:
Enter sex of respondent
!CAT
1 Male
2 Female}
{
!GEN,M,TITLE,SEX}
{TITLE:
!CAT,A
1 Mr.
2 Ms.}
{
Now we can address you by your title,\:TITLE:
\:NAME:
!DISP,1}
```

In the above example the response entered in SEX is moved to TITLE. Question TITLE, which doesn't show on the screen, associates new text (Mr. or Ms.) corresponding to the original response (Male or Female), and can be back-referenced in later questions.

Generating Data for CAT,C and CAT,D Questions

The allowable responses to a CAT question subtype C or D are determined by the responses to a previous question. You must first use a GEN,M or GEN,O statement to copy the data from the first question into the second in order to control which responses will be allowed.

For example, you may have previously asked:

```
WHICH HOSPITALS IN THE AREA ARE YOU AWARE OF?
```

And now you are asking:

```
OF THOSE YOU MENTIONED, WHICH IS YOUR FAVORITE?
```

By specifying this second question as a CAT,D question, and using a GEN,M statement to move the data from the first question into it, you would display and allow only responses from the first question.

Here is a sample GEN,M/CAT,D question sequence:

```
EX:
{KNOWHOSP:
WHICH HOSPITALS IN THE AREA ARE YOU AWARE OF?
!CAT,,3
1 SHRINERS HOSPITAL
2 ST. HELEN'S HOSPITAL
3 COUNTY HOSPITAL}
{
!GEN,M,FAVOHOSP,KNOWHOSP}
{FAVOHOSP:
OF THOSE YOU MENTIONED, WHICH IS YOUR FAVORITE?
!CAT,D
[SAMEAS KNOWHOSP]}
```

The first question is asked. The GEN,M statement then moves the data from KNOWHOSP to FAVOHOSP. Only responses that were given in the first question will be displayed in the response list for the second question. After the second question is asked, any responses not mentioned are automatically deleted from the second question's data.

You may also need to do the opposite — have only those responses that were not mentioned in the first question listed as possible responses in a later question. Using the above example, the first CAT question and the GEN,M statement would remain the same. However, the last question would be a CAT,C question. You will need to follow it with a GEN,T if you want to remove the answers previously entered.

Here is a sample GEN,M/CAT,C /GEN,T question sequence:

```
EX:
{KNOWHOSP:
Which hospitals in the area are you aware of?
!CAT,,3
1 SHRINERS HOSPITAL
2 ST. HELEN'S HOSPITAL
3 COUNTY HOSPITAL
(-)4 DK/NA}
{
!IF NUMITEMS(KNOWHOSP)=3
!GOTO,LAST}
{
!GEN,M,OTHRHOSP,KNOWHOSP}
{
!GEN,Z,OTHRHOSP,4}
{OTHRHOSP:
Are you aware of any of these other hospitals in
your area?
(READ LIST)
!CAT,C,3
[SAMEAS KNOWHOSP]}
{
!IF NOT(KNOWHOSP(4))
!GEN,T,OTHRHOSP,KNOWHOSP}
{LAST:
!GOTO}
```

For the question KNOWHOSP, the DK/NA is marked exclusive (-). If the respondent was aware of all 3 named hospitals, they will skip around the GENs and CAT,C question. Then the data is moved from KNOWHOSP to OTHRHOSP with the DK/NA response removed (so that it is an allowable answer again for OTHRHOSP).

For the question OTHRHOSP, only hospitals that were not mentioned in response to the question KNOWHOSP will be displayed in the response list. CAT,C questions do not automatically remove the data brought in by the GEN,M statement. This is because you may want a net of all responses saved. If you want to remove the responses brought in from the first question, you would use a GEN,T statement.

The GEN,T statement will take all responses out of question OTHRHOSP that also exist in question KNOWHOSP. Only those responses unique to OTHRHOSP will remain. This is done under the condition of the first question's response not being DK/NA. If you neglect to put this condition in, you may end up removing a legitimate DK/NA response to the OTHRHOSP question.

Use of the header statement option FLAG_DISALLOWED_CATS will show all responses but will mark the disallowed ones as not available for this question (CAT,C or D). It will not affect what can be entered into the data.

NOTE: Use of the CAT,I or FLD,I question types is a better way to handle these situations.

USING THE ALIAS OPTION

The ALIAS option on the question label line is used when you have the same question in several different places in the questionnaire and you later want to reference the answer, wherever it may be.

You can't have two questions with the same label, but you can "alias" two (or more) questions together so that when you back-reference the one name, you get the answer even if it was originally asked under a different question name.

For example, males and females might have been asked a different series of questions, but in the middle of the series, the same question was asked of each. After everyone is asked the same questions again, you want to back-reference the answer they gave, but don't want to have to make two questions (based on gender) or have two separate back-references in one question.

Here is what you could do:

```
EX: (male series)
{dogs:
How many dogs do you have?
!NUM,,,0-10,,dk}
(female series)
{fdogs: [dogs] ALIAS=dogs
How many dogs do you have?
 !NUM,,,0-10,,dk}
 (all together now)
 {Activity:
 !IF dogs>0
 What activities do you share with your \:dogs:
 dog(s)?
 !CAT,,6
 01 Walking
 02 Running
 03 Swimming
 04 Frisbee
 05 Catch/Fetch
 06 Napping}
```

If all you wanted to do was display the number of dogs on the screen, the ALIAS=dogs would have been sufficient on the FDOGS question. Since the IF question on the ACTIVITY question needed to know the answer as well, we made sure the same data location was used for both questions (thus the [dogs] on the FDOGS question).

Unless the order of the answers is important to you, a better way to handle this is the following:

```
{icecream:
What flavor of ice cream is your favorite?
!CAT,,1
01 Chocolate
02 Strawberry
03 Vanilla
04 Neapolitan
05 Cookies and Cream
06 Tutti Frutti
07 Other (specify)
(-) 08 None}
{otherice:
!IF ICECREAM(07)
Enter other flavors:
!VAR,,20,3}
{
!GEN,M,ICECRM2,ICECREAM}
{icecrm2: [icecream] ALIAS=ICECREAM
What flavor of ice cream is your favorite?
!CAT,A,1
01 Chocolate
02 Strawberry
03 Vanilla
04 Neapolitan
05 Cookies and Cream
06 Tutti Frutti
07 \:OTHERICE:
(-) 08 None}
!IF ICECREAM(08)
!GOTO,SKIPWHY}
{WHY:
Why is \:ICECREAM your favorite?
!TEX}
{SKIPWHY:
!GOTO }
```

This way, you avoid the printing of “Other (specify)” as well as the other ice cream.

INDEX