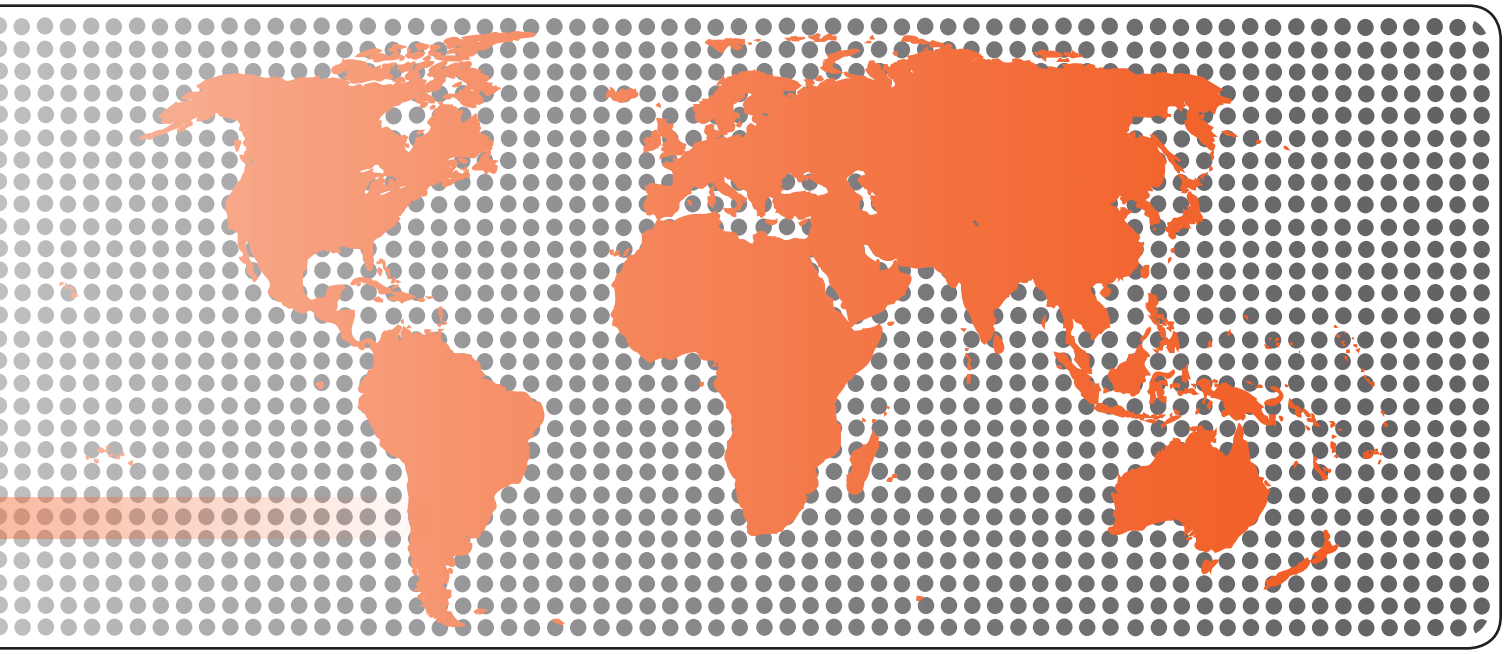
GTSS Global Adult Tobacco Survey (GATS)



Programmer's Guide to General Survey System

Global Adult Tobacco Survey (GATS)
Programmer's Guide to General Survey System

Version 2.0

November 2010

Global Adult Tobacco Survey (GATS)
Comprehensive Standard Protocol

GATS Questionnaire
Core Questionnaire with Optional Questions
Question by Question Specifications

GATS Sample Design
Sample Design Manual
Sample Weights Manual

GATS Fieldwork Implementation
Field Interviewer Manual
Field Supervisor Manual
Mapping and Listing Manual

GATS Data Management
Programmer's Guide to General Survey System
Core Questionnaire Programming Specifications
Data Management Implementation Plan
Data Management Training Guide

GATS Quality Assurance: Guidelines and Documentation

GATS Analysis and Reporting Package
Fact Sheet Template
Country Report: Tabulation Plan and Guidelines
Indicator Definitions

GATS Data Release and Dissemination
Data Release Policy
Data Dissemination: Guidance for the Initial Release of the Data

Tobacco Questions for Surveys: A Subset of Key Questions from the Global Adult Tobacco Survey (GATS)

Suggested Citation

Global Adult Tobacco Survey Collaborative Group. *Global Adult Tobacco Survey (GATS): Programmer's Guide to General Survey System, Version 2.0*. Atlanta, GA: Centers for Disease Control and Prevention, 2010.

Acknowledgements

GATS Collaborating Organizations

- Centers for Disease Control and Prevention
- CDC Foundation
- Johns Hopkins Bloomberg School of Public Health
- RTI International
- University of North Carolina Gillings School of Public Health
- World Health Organization

Financial Support

Financial support is provided by the Bloomberg Initiative to Reduce Tobacco Use, a program of Bloomberg Philanthropies, through the CDC Foundation.

Disclaimer: The views expressed in this manual are not necessarily those of the GATS collaborating organizations.

Contents

Chapter **Page**

1. Introduction 1-1
- 1.1 Overview of the Global Adult Tobacco Survey 1-1
- 1.2 Use of This Manual 1-2
- 1.3 Hardware and Software System Requirements 1-5
- 1.4 Other Software and Hardware Requirements 1-6
- 1.5 GSS Program Files 1-6

2. Case Management and Transmission System 2-1
- 2.1 Case Management 2-2
- 2.2 Transmission Program (Xmit) 2-9
- 2.3 Updater Program 2-11

3. Web Management Tools 3-1
- 3.1 Case Management Options 3-2
- 3.2 Certification 3-2
- 3.3 Reports 3-3
- 3.4 User Maintenance 3-3

4. General Survey System Engine 4-1
- 4.1 Overall Design 4-1
- 4.2 Example 1—Program a Small Questionnaire 4-3
- 4.3 Example 2—Using GSS to Program the GATS Household Questionnaire 4-16

5. GSS Developer's Tool Set 5-1
- 5.1 GSS Developer's Tool Set 5-1
- 5.2 Questionnaire Designer 5-11
- 5.3 Data Aggregation Software 5-20
- 5.4 Aggregate Data—File Builder Tool 5-24

6. Files and Structures 6-1

7. How to Configure the GSS for a Handheld 7-1
- 7.1 How to Configure a Pocket PC 7-1
- 7.2 GSS Files 7-2
- 7.3 How to Configure the Case Management System 7-3
- 7.4 How to Configure the GSS 7-8
- 7.5 How to Configure the Transmit Program 7-10
- 7.6 How to Configure the Updater Program 7-10
- 7.7 Language Customization 7-10

Appendix A. GSS Programming Language A-1

Appendix B. Review of the Main Data Tables B-1

Appendix C. Frequently Asked Questions C-1

Appendix D. List of Acronyms D-1

Appendix E. GSS Installation into an iPAQ Model 2490 ..... E-1

Appendix F. GSS Installation into an iPAQ Model 210 ..... F-1

Exhibits

Number		Page
Exhibit 1-1.	Sample Configuration Using Fully Networked Model	1-4
Exhibit 2-1.	Description of Transmitted Tables	2-11
Exhibit 4-1.	GSS Engine Screen Elements	4-3
Exhibit 4-2.	GSS Engine Screen Types	4-4
Exhibit 4-3.	GATS Individual Questionnaire Subset	4-5
Exhibit 4-4.	Screen Information: Questions Table	4-8
Exhibit 4-4a.	Question Text: The Texts Table	4-8
Exhibit 4-5.	Answer Text and Codes: Answers Table	4-9
Exhibit 4-6.	Initialization Data: INI Table for Survey Form	4-10
Exhibit 4-7.	GSS Engine Special Instructions Options	4-11
Exhibit 4-8.	GSS Screen Properties	4-12
Exhibit 4-9.	Household Questionnaire	4-17
Exhibit 4-10.	GSS Implementation of the GATS Household Questionnaire: Questions Table	4-23
Exhibit 4-11.	GSS Implementation of the GATS Household Questionnaire: Answers Table	4-27
Exhibit 5-1.	Start Screen of Main Menu for GSS Developer Tools	5-2
Exhibit 5-2.	CMS Grid Designer: First Screen	5-9
Exhibit 5-3.	Designer Main Screen	5-13
Exhibit 5-4.	Import Layouts	5-16
Exhibit 5-5.	Aggregation Diagram	5-26
Exhibit 5-6.	Aggregation Summary Report (file Aggrpt.htm)	5-32
Exhibit 5-7.	Status Report (Statrpt.htm)	5-34
Exhibit 5-8.	GSS Variable List Utility	5-38
Exhibit 5-9.	Request for Path for Storing Variable List (.txt) File	5-39
Exhibit 5-10.	Request for Path for Storing Output Data File (Flat File or CSV Delimited File)	5-40
Exhibit 5-11.	Request for Path to Input SDF File with Raw Data	5-41
Exhibit 5-12.	Transpose Summary Screen	5-42
Exhibit 5-12a.	Reporting Utility	5-43
Exhibit 5-13.	Survey Event Status Report HQ Case Status	5-44
Exhibit 5-14.	Filter Screen	5-45
Exhibit 5-15.	Questionnaire Frequency Distributions	5-45
Exhibit 5-16.	FI Exceptions Cases with No Final Codes and >= Four Result Codes	5-46
Exhibit 6-1.	Purpose of Internal Tables of CMSDB.sdf	6-2
Exhibit 6-2.	CMS Database Table: AddressLOG	6-3
Exhibit 6-3.	CMS Database Table: Delrecs	6-4

Exhibit 6-4. CMS Database Table: DU 6-5
Exhibit 6-5. CMS Database Table: DUEvt 6-6
Exhibit 6-6. CMS Database Table: EvtDef 6-7
Exhibit 6-7. CMS Database Table: FSTable 6-7
Exhibit 6-8. CMS Database Table: INI 6-7
Exhibit 6-9. CMS Database Table: InterviewInfo 6-8
Exhibit 6-10. CMS Database Table: Messages 6-9
Exhibit 6-11. CMS Database Table: Notes 6-9
Exhibit 6-12. CMS Database Table: Responses 6-10
Exhibit 6-13. CMS Database Table: Tasks 6-11
Exhibit 6-14. CMS Database Table: TrainingCases 6-11
Exhibit 6-15. CMS Database Table: VERINFO 6-12
Exhibit 6-16. Purpose of Questionnaire Database Tables 6-13
Exhibit 6-17. User Maintainable Database Tables for a Survey Form: Answers 6-14
Exhibit 6-18. User Maintainable Database Tables for a Survey Form: INI 6-14
Exhibit 6-19. User Maintainable Database Tables for a Survey Form: Questions 6-15
Exhibit 6-20. User Maintainable Database Tables for a Survey Form 6-16
Exhibit 6-21. Static Database Tables for a Survey Form, Not to Be Modified by the User: FormatType 6-17
Exhibit 6-22. Static Database Tables for a Survey Form, Not to Be Modified by the User: Opers 6-17
Exhibit 6-23. Static Database Tables for a Survey Form, Not to Be Modified by the User: Types 6-17
Exhibit 6-24. Optional Database Tables for a Survey Form: AnswerKeys 6-17
Exhibit 6-25. GATS_DBUtil Start Screen 6-18
Exhibit 6-26. GATS_DBUtil after a Verify of CMS 6-19
Exhibit 6-27. Gats_DBUtil Options Available After a Successful Verify 6-20
Exhibit 7-1. CMS Initialization Table, the INI Table in ProjectName_CMSDB.sdf 7-5
Exhibit 7-2. Data Fields of the Cases Table (CMSDB DU) 7-7
Exhibit 7-3. List of Fields for GATS_CASEFILE.txt 7-8
Exhibit 7-4. GSS Initialization Table, the INI Table 7-9
Exhibit 7-5. Messages Table 7-11

1. Introduction

Tobacco use is a major preventable cause of premature death and disease worldwide. Approximately 5.4 million people die each year due to tobacco-related illnesses – a figure expected to increase to more than 8 million a year by 2030. If current trends continue, tobacco use may kill a billion people by the end of this century. It is estimated that more than three quarters of these deaths will be in low- and middle-income countries[1]. An efficient and systematic surveillance mechanism is essential to monitor and manage the epidemic.

The ***Global Adult Tobacco Survey*** (GATS), a component of Global Tobacco Surveillance System (GTSS), is a global standard for systematically monitoring adult tobacco use and tracking key tobacco control indicators. GATS is a nationally representative household survey of adults 15 years of age or older using a standard core questionnaire, sample design, and data collection and management procedures that were reviewed and approved by international experts. GATS is intended to enhance the capacity of countries to design, implement and evaluate tobacco control interventions.

In order to maximize the efficiency of the data collected from GATS, a series of manuals has been created. These manuals are designed to provide countries with standard requirements as well as several recommendations on the design and implementation of the survey in every step of the GATS process. They are also designed to offer guidance on how a particular country might adjust features of the GATS protocol in order to maximize the utility of the data within the country. In order to maintain consistency and comparability across countries, following the standard protocol is strongly encouraged.

GATS manuals provide systematic guidance on the design and implementation of the survey.

1.1 Overview of the Global Adult Tobacco Survey

GATS is designed to produce national and sub-national estimates among adults across countries. The target population includes all non-institutionalized men and women 15 years of age or older who consider the country to be their usual place of residence. All members of the target population will be sampled from the household that is their usual place of residence.

GATS uses a geographically clustered multistage sampling methodology to identify the specific households that Field Interviewers will contact. First, a country is divided into Primary Sampling Units, segments within these Primary Sampling Units, and households within the segments. Then, a random sample of households is selected to participate in GATS.

The GATS interview consists of two parts: the *Household Questionnaire* and the *Individual Questionnaire*. The *Household Questionnaire* (household screening) and the *Individual Questionnaire* (individual interview) will be conducted using an electronic data collection device.

The GATS interview is composed of two parts: *Household Questionnaire* and *Individual Questionnaire*. These questionnaires are administered using an electronic data collection device.

[1] Mathers, C.D., and Loncar D. Projections of Global Mortality and Burden of Disease from 2002 to 2030. *PLoS Medicine*, 2006, 3(11):e442.

At each address in the sample, Field Interviewers will administer the *Household Questionnaire* to one adult who resides in the household. The purposes of the *Household Questionnaire* are to determine if the selected household meets GATS eligibility requirements and to make a list, or roster, of all eligible members of the household. Once a roster of eligible residents of the household is completed, one individual will be randomly selected to complete the *Individual Questionnaire*. The *Individual Questionnaire* asks questions about background characteristics; tobacco smoking; smokeless tobacco; cessation; secondhand smoke; economics; media; and knowledge, attitudes, and perceptions about tobacco.

1.2 Use of This Manual

The purpose of this document is to provide an overview of the General Survey System (GSS) and to help users understand the suite of software tools that make up the GSS. Information in this document includes an overview of the technical architecture, the business architecture, and the high-level requirements for the system. Detailed architecture and design models, or references to them, may also be included where appropriate. The document is intended for information technology staff who will support the GSS and for individuals who will program forms within the GSS.

The GSS is a suite of software tools developed to facilitate the administration, collection, and management of survey data on handheld computers, specifically a Microsoft Windows-based platform running Windows Mobile 5.0 or Mobile 6.0, often called Pocket PC systems. The software system is designed to support field data collection activities where Field Interviewers collect data using handheld computers. The systems have been developed and tested using Hewlett Packard (HP) iPAQ Pocket PC systems[2]. The software consists of six main programs, each dedicated to a specific function:

1. **CMS**: a case management system (CMS) that allows users to manage the case load on the Pocket PC.
2. **GSS Engine**: a questionnaire presentation engine that allows execution of data collection forms or questionnaires on the Pocket PC.
3. **Xmit**: a data transmission program that allows bidirectional movement of data, program updates, and control information to and from the Pocket PC over dialup, wireless, or wired Ethernet.
4. **Developer's Tool Set**: a developer's menu system that organizes the access to the PC-based components of GSS.
5. **Designer**: a questionnaire design program that provides a visual interface for preparing and/or modifying a survey instrument. The Designer allows the creation, deletion, and modification of questions in two languages at a time in the GSS.
6. **Project Web site**: a Web-based suite of tools that facilitates survey management, survey monitoring, and reporting, and brokers the data transmissions to and from the Pocket PC to back-end database servers.

[2] GATS uses General Survey System (GSS) software, which is designed to run on a Windows Mobile platform and has been tested and implemented using a Hewlett-Packard (HP) iPAQ© handheld PDA computer. Use of "iPAQ" is for identification only and does not imply endorsement by any of the GATS collaborating organizations.

Exhibit 1-1 provides a diagram of the system programs and how they interact. Three of the major programs—CMS, GSS, and Xmit—run on the Pocket PC that the Field Interviewer uses. The developer's tool set runs on a desktop PC. The Project Web site runs on a centrally based Microsoft IIS Web Server running ASP.net Web pages linked to a Microsoft SQL Server database for data storage. **Exhibit 1-1** is a sample configuration diagram using a fully networked model.

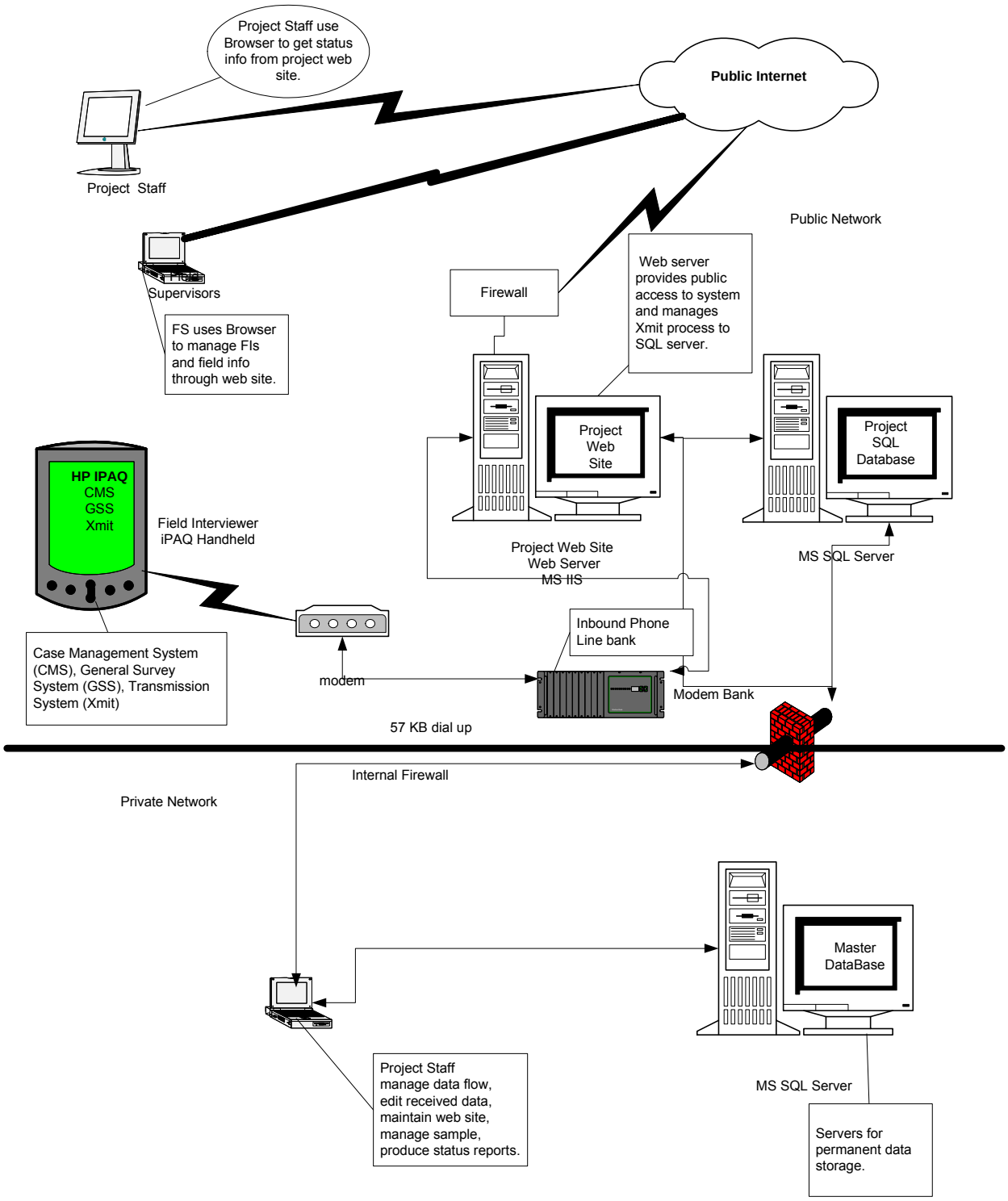
The exhibit shows the Field Interviewer link to the central systems as a dialup modem, but other network connections to the Internet, including wireless and wired Ethernet, are supported for communication between the iPAQ and the central systems. The network communication link allows the Field Interviewer to do the following:

- Receive new cases.
- Return cases for reassignment to other Field Interviewers.
- Upload data collected during the day via nightly or other periodic transmissions, including:
  - questionnaire answers,
  - status codes of work in progress,
  - case notes, and
  - address corrections.
- Receive updates to software and database tables (questionnaires only).

In addition, a path is shown for project staff and Field Supervisor staff to communicate with the systems via a Web site. This set of Web-based software tools allows the supervisors, project staff, and clients to use a standard browser to access the Web site. The Web site is password protected, and access to various functions is determined by the user's role on the project. Depending on the role assigned to a user, some functions will be enabled while others will not. Following are some functions that users can perform:

- Obtain status and management reports.
- Certify Field Interviewers as qualified to receive cases.
- Assign cases to and recall cases from Field Interviewers.
- Execute quality control procedures, such as the following:
  - Review data collected.
  - Check on transmission status.

Exhibit 1-1. Sample Configuration Using Fully Networked Model



1.3 Hardware and Software System Requirements

The hardware platforms and software requirements for the programs are as follows:

- **Pocket PC—GSS, CMS, and Xmit Programs**
  - Hardware
    - Hewlett Packard Pocket PC
    - Models: iPAQ hx2490c, iPAQ 110/210/211/212
  - Software
    - Operating system Windows Mobile 5 or 6
    - Microsoft.NET Compact Framework
    - SQL Server Compact Edition runtime libraries—Microsoft's embedded database SQL
- **Desktop PC—Developer's Tool Set**
  - Hardware
    - Standard desktop PC
  - Software
    - Operating system Windows XP with Service Pack 3 or Windows 7
    - MS.Net Framework
    - MS Active Synch 4.5
    - SQL Server Compact Edition—Microsoft's embedded database SQL
- **Web and Database Servers—Xmit and Web Management Programs**
  - Project Web site operating system and development tools
    - Microsoft IIS (Web server operating system)
    - Microsoft ASP.net (development tools)
  - Database server
    - Microsoft SQL Server 2000 or later

An installation disk is available for these tools and the required Microsoft products that do not require licenses.

1.4 Other Software and Hardware Requirements

In addition to the software listed above, users may need the following desktop software:

- **Secure FTP client.** This is optional software that can be used for transmitting data files between data collection sites to a central location.
- **Encryption software program (e.g., PGP).** For security purposes, users should encrypt data files containing sensitive information prior to saving to an external location. Any commercially available software that meets the National Institute of Standards and Technology (NIST) encryption standards may be used to encrypt the data. There are also open-source encryption tools that will provide adequate file security.
- **Spreadsheet application (e.g., Excel, Quattro-pro).** Users may need to create spreadsheets to summarize recruitment outcomes and track any data issues.
- **Secure Digital (SD) memory card readers.** Desktop PCs used for data consolidation should have SD memory card readers to allow the reading of memory cards from Field Interviewer Pocket PCs. These are inexpensive USB devices that are widely available.
- **Microsoft Access** version 2003 or later.
- **Microsoft Office** version 2003 or later.

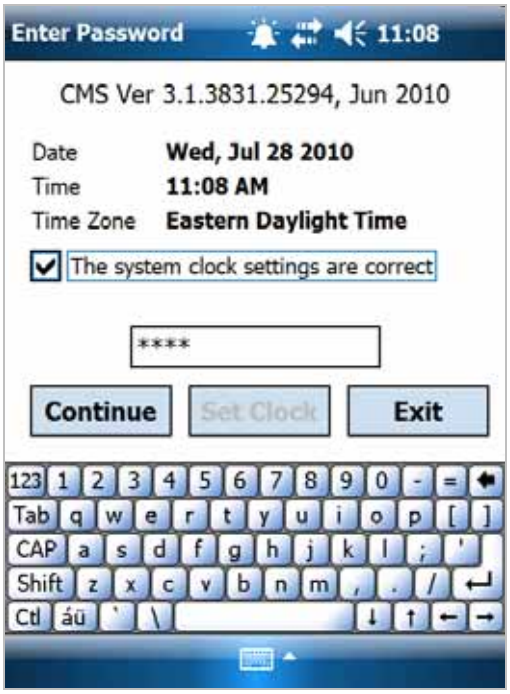
1.5 GSS Program Files

Each site will receive copies of the files and an installation program for the following:

- Handheld data collection applications, a *Household Questionnaire*, and an *Individual Questionnaire* in the native country language(s) and in English for collecting interview data specific to the country's questionnaire.
- Developer's Tool Set containing PC-based programs for creating/adding/modifying local questions in the GSS, managing GSS databases, building required support files, and viewing GSS data files. Because changes to the questionnaire could alter the system specifications (e.g., variable names, response values and formats, consistency checks), the questionnaire cannot be modified without the approval of the GATS Questionnaire Committee.
- iPAQ-based tools to configure iPAQs and load the CMS, GSS tools (and their required support files) on iPAQs.

2. Case Management and Transmission System

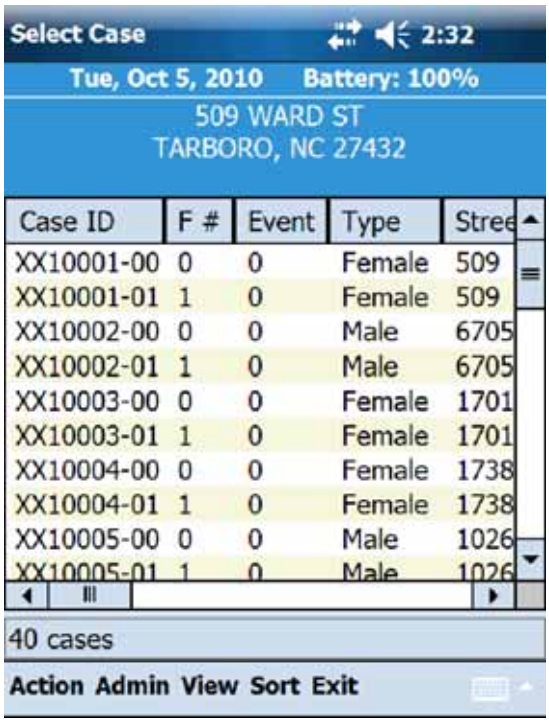
GSS users (typically Field Interviewers) see the CMS as the first iPAQ-based software module they use. It is the software interface to the GSS engine and, hence, to data capture. The CMS is used to manage the workload of cases on the Pocket PC. It is a password-protected system. Users must enter a login password to enter the system each time the CMS is started. During logon, passwords are not displayed; they are hidden by masking characters. The system keeps track of failed login attempts and notifies the user with messages at specified intervals. After five failed attempts, the CMS is locked and cannot be started without an unlock code, which has to be supplied by supervisory staff. The code is available from utilities on a Web site or from printed lists showing the code for each day of the year. The Field Interviewer must contact the Field Supervisor to obtain this unlock code, which is valid only for the day it is issued. The CMS login screen is shown below.



The CMS functions are described briefly in the next section (see the ***GATS Field Interviewer Manual*** and ***GATS Field Supervisor Manual*** for detailed screenshots and descriptions of how to use the CMS).

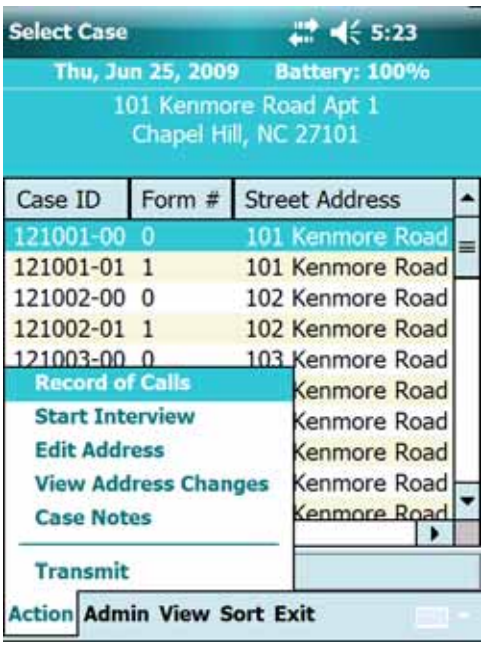
2.1 Case Management

When the CMS begins, the program shows all active cases available to the Field Interviewer on the Pocket PC. The list shows all necessary information for a Field Interviewer to locate each household and to conduct field work. From this list of households, Field Interviewers can plan strategies to finish their assignments. Several tools are available to help Field Interviewers organize the households in their assignment. The CMS Select Case screen (case grid) can be sorted in different orders using the variables in case grid such as Case ID, Street, Event Code, City, State, Zip Code, County, and so forth. The list can also be filtered to display only certain types of cases (pending, completed, etc.). The CMS main screen is shown below loaded with a set of 40 training cases.

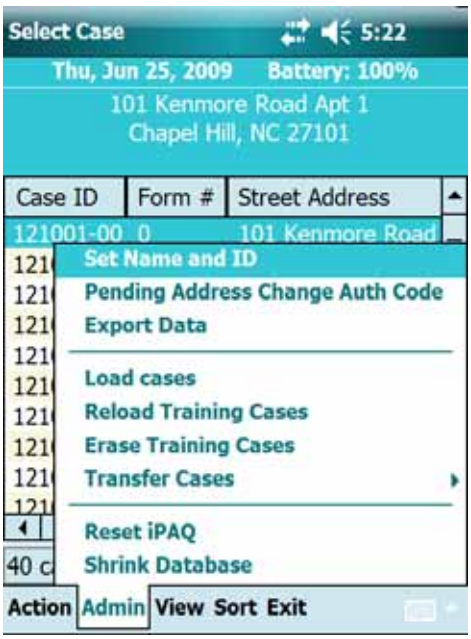


The purpose of the CMS is to allow Field Interviewers to manage, view, and transmit their household assignment. The following screens show four major menus with their options on the Select Case screen that facilitate those goals. These screen show the option Transfer Cases in the Admin menu and Select Items menu in the View menu items. The default option in GATS is that these options are not used and they will not show in menus.

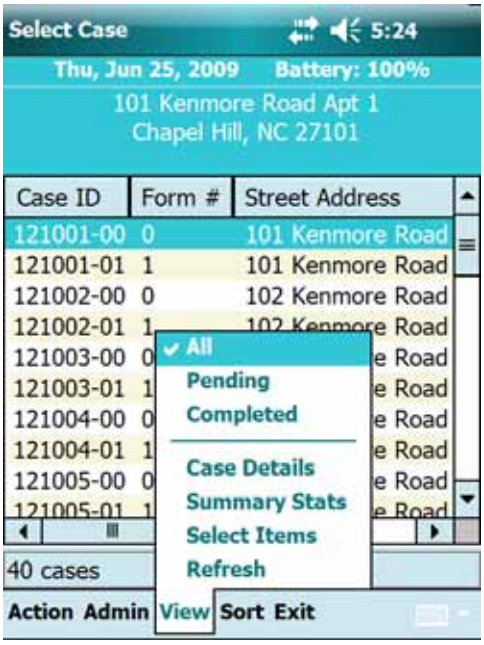
Action Menu



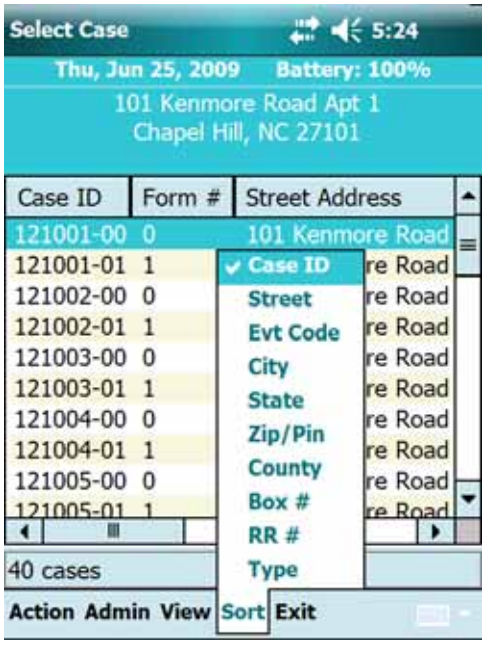
Admin Menu



View Menu



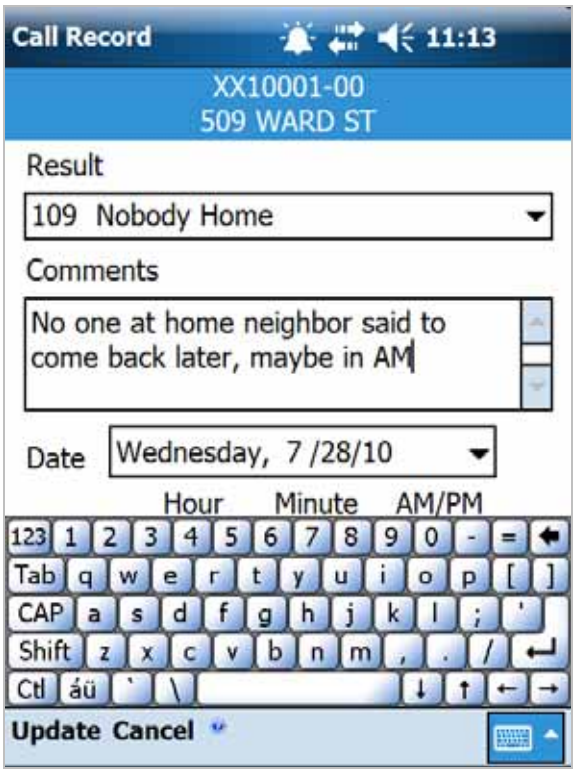
Sort Menu



Action Menu

Record of Calls

As they perform field work on a case, Field Interviewers can enter result codes (for a given case) that document the work performed. The system allows Field Interviews to enter result codes from a list of available events. The result codes for a given project are table driven and reasonably easy to adapt from project to project, although rules or validations based on result codes may require CMS code changes to implement. The cases in the grid are color coded according to the most recent result code. Cases that have not been worked (result code 0) are colored black, cases that are pending are colored green, and cases that are complete are colored red. The definition of pending and complete is controlled by INI parameters (INI table in the CMSDB database) that list the result codes associated with a given state.
A Field Interviewer can also enter comments and specify the date and time of an event. A sample Record of Calls screen is shown below for a case where the Field Interviewer found no one home.



When a Field Interviewer enters a result code by tapping the Update menu item, the system goes through a series of checks to validate the event. If the event fails to validate, the system prompts the Field Interviewer to take appropriate actions. The major validations are as follows:

- Some events require predecessors with certain conditions. If the conditions are not met, such an event cannot be accepted. The specific rules are updated for a given project.
- Some events require authorization codes. If the code is missing, such an event cannot be accepted.
- Some events will trigger other events. This is handled automatically in the system.

These rules or validations may require CMS code changes from project to project.

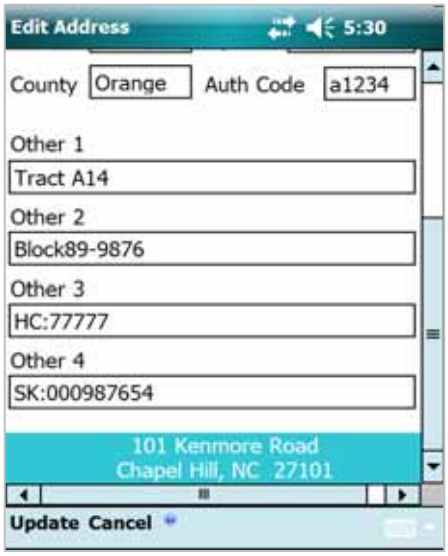
If a Field Interviewer edits or deletes an event, the system reverses some of the steps described above. After a Field Interviewer enters an event and **before the event** is transmitted to the central servers, the Field Interviewer has the option to edit or delete the event. Once an event has been sent to the central servers, it can no longer be edited. See the ***GATS Field Interviewer Manual*** and ***GATS Field Supervisor Manual*** for details about GATS result code use and protocol.

Start Interview

On the Select Case screen, Field Interviewers can select a questionnaire and start the interview using the Action Menu. The interview consists of a series of questions, and at the end of the interview result codes may be assigned based on the results of the interview. The interview data and all the CMS data are stored in Windows Mobile Compact Edition SQL databases on the iPAQ. In the GATS project the database file is the GATS_CMSDB.sdf file.

Edit Address

If a Field Interviewer discovers that the address of any household shown on the iPAQ is incorrect, he or she can edit the address. The Street Number, Apartment Number, Box Number, RR Number, Street Name, City, State, County, and Zip Code fields are available for editing. These fields' labels can be renamed by editing the GSS messages table to be more appropriate for a given country's locator standards. Any address changes for a household require an authorization code to be transmitted to the central servers. Field Interviewers can enter the authorization code on the Edit Address screen or enter it later. The default in GATS is to provide a generic authorization code so that Field Interviewers do not have to get one for changes. An example Edit Address screen is shown in the following two screenshots that show the top part and bottom part of the screen. The items Other 1 through Other 4 are data items, passed from the Case file that may be used to drive questionnaire routing or sample selection. These are not used in GATS.



View Address Changes

Users can view changes to an address by using this menu option. First, select a case, tap action, and then View Address Changes. This will display the screen below that shows both the old and new addresses for the selected case.

Case Notes

The Case Notes menu item allows Field Interviewers to maintain notes about a case in a diary, free text form. These notes are saved in the database in the Notes table and transmitted along with all the other data. To make a case note or update a case note:

Step 1: **Select a case** in the case grid.

Step 2: Tap **Action**.

Step 3: Tap **Case Notes**.

A screen displays that allows text entry in free form for the selected case. The screenshot below shows a sample Case Note that has been made for a selected case.

Transmit—Data Transmission

From the Select Case screen, Field Interviewers can start the transmission of all data collected in the iPAQ since the last transmission (**Section 2.2** describes the operations of the transmission system). The dataset includes, if applicable, Events, Interview Results, Edited Addresses, and Case Notes. The transmission systems require an active network link between the iPAQ and a network. This can be a direct dial connection, over a modem, to a dedicated network system or a wireless or wired Ethernet connection to the public Internet. A Field Interviewer can connect the iPAQ directly to a modem and phone line and begin the transmission. Transmitted data are immediately available in the SQL Server databases at the central site.

Admin Menu

Set Name and ID—System Configuration

When a user receives a new iPAQ, the machine must be configured by providing information such as the Field Interviewer ID, Field Interviewer Name, Dialup User Name, and Dialup Password. Until the system is configured, users cannot perform any other functions in the system. These items may be managed through the Admin menu item Set Name and ID.

Pending Address Change Auth Code

This CMS feature is not used in GATS.

Export Data

If network-based communication is not available, the CMS also supports moving data into and out of the iPAQ via memory cards—the SD memory cards that are supported in the iPAQ. The memory cards can be used to load initial cases, update cases, remove or inactivate cases (to allow transfer to a different Field Interviewer), and export data for aggregation across Field Interviewers. The Load Cases submenu

item in the Admin menu adds or removes cases, while the Export Data submenu item creates labeled snapshots of the database in an Export folder on the SD card. Once the snapshot is made, the card can be removed and the export files used to aggregate or back up data in a number of ways. The optional Transfer Cases menu item, if the option is turned on, allows users to create a transfer file on the iPAQ's SD card to transfer a selected number of their cases to another iPAQ.

Load Cases

The Load Cases feature causes the CMS to look for a file in the root folder of the iPAQ SD card named GATS_CaseFile.txt. If this file exists, it will read it and load cases targeted for a given Field Interviewer into the CMS case grid. The key for loading cases is the Field Interviewer ID (FI ID). This is the FI ID displayed in the Set Name and ID menu. That is, all cases in GATS_CaseFile.txt that have the a matching Field Interviewer ID(FI ID) as the on set on the iPAQ will be loaded; all others will be ignored.

Reload or Erase Training Cases

On the iPAQ, sample or mock cases are available for training. Field Interviewers can load these cases by using the Reload Training Cases menu option. Training cases can be retrieved or removed at any time. Once loaded, a training case can be treated as a normal case, thus allowing practice use of the system, including transmission. If the training cases are erased, via the Erase Training Case menu option, all associated data are also deleted. Training cases have restricted ID numbers that start with XX to differentiate them from live cases. No live IDs that start with XX should ever be created. Data managers should be prepared to see and ignore cases that have CaseIDs that start with XX in the GSS data tables. Users can edit the training case table (TrainingCases in the GATS_CMSDB file) to make the training cases look like live cases in their environment).

IPAQ Reset and Shrink Database

The CMS program provides a function to reset the machine to refresh memory for consistent performance. The iPAQ reset functions like a reboot on a standard Windows PC. In addition, the Shrink Database menu option allows maintenance on the internal database. It instructs the iPAQ to clean and compress the database.

Purge Data Function

The purge function is a configuration option of the CMS and is not used in GATS. For projects that work in multiple data collection cycles (waves) or have fixed periods of data collection, it is useful to be able to delete the old period's work before a new survey wave begins. At the end of a work period, the Field Interviewer will purge the period's project data before starting work on the next period. The purge function ensures all data have been transmitted to the Central Office before removing any data from the iPAQ. This function also requires a code from the Field Supervisor to be performed.

View and Sort Menus

The View Menu items allow users to call up different views of the data and displays of additional information. The view items (All, Pending, Completed) controls which rows are displayed in the case grid and the sort items controls the sort order of the selected rows. Data are not added or deleted using these options; they control only how data are displayed in the grid. You can control the view to show all cases, only pending cases, or only completed cases. The definitions of pending and completed are INI parameters that users can specify. GATS definitions are shown below:

VariableName	VariableValue	Comment
EvtcodeComplete	200, 400	Result codes that are complete quex
EvtCodeFinal	200, 201, 202, 203, 204, 205, 206, 208,209, 999, 400, 401,402, 403, 404, 407, 408, 409	Result codes that are final status codes
EvtCodePending	0,102, 103, 104, 105, 106, 108, 109, 302, 303, 304, 307, 308, 309,887	Result codes that are pending status codes
EvtCodeScreened	200	Result code indicating screening complete and one person selected for more data collection

Users can call up a view that displays selected details of the case and also request a summary statistics view that displays a summary of the statuses of the cases. The Refresh option refreshes the Case Grid.

The Sort menu allows the users to control the sort order of the cases in the Select Screen. Cases can be sorted by a number of case variables (e.g., CaseID, Street, County, Zip/Pin, etc.). The default sort order is by CaseID.

2.2 Transmission Program (Xmit)

The Transmission program allows data to move back and forth between the iPAQ and the host server(s) (see the ***GATS Field Interviewer Manual*** for detailed screenshots and descriptions of how to use the Xmit program).

The features of the Transmission program are described below:

- Configure
  - The Configure screen allows the user to edit the dialup username and password. The transmission timeouts can also be modified if the user has problems with slow transmissions. Under most circumstances, these settings do not need to be changed.
- Transmit
  - The users should press the Transmit button to begin the transmission. If a connection exists on the device, it is used for the transmission. Otherwise, the system attempts to use the dialup connection currently selected in the Connection dropdown menu. During the transmission, the current status displays on the screen as well as a log of events during transmission. A transmission session uploads all untransmitted data from the SQL CE database tables in the iPAQ to the Central Office. It checks for new information to be downloaded from the servers. This could be new cases to work or requests to return cases back to the Central Office for reassignment to other staff. In addition, an option allows updates to the iPAQ SQL CE data table to update the questionnaire application (new question texts, answer texts, etc.).
- Save log
  - After a transmission is complete, the user can save the log of the transmission. The log file contains detailed information about the transmission, including the version number of the transmission application, the number of records sent/received from each table, and any errors that occurred. The log file will be automatically saved in a subfolder of the application

with the date and time appended to the file name. We recommend that the user save the log file. If an error occurs in transmission, the log file will help track and solve the problem.

- Miscellaneous items
  - The Xmit program is written in Visual Basic.NET using the Microsoft Compact Framework libraries.
  - The program uses the SQL Server CE database, a compact version of Microsoft SQL Server. The SQL CE database is a shared database, with multiple data tables used by all of the iPAQ software programs. It is stored as an encrypted database file.

Transmission Process

All data transmissions are performed using the HTTPS protocol over TCP/IP. Web services are used on the server to facilitate communication with the iPAQ. The transmission process also relies on a series of acknowledgments for successful transmission.

Data are transmitted on a per-table basis, except for the responses table, which sends data on a per-case basis to limit the size of the data blocks to reduce transmission errors. For each table, use the following process to transmit data:

- Transmitting records from server to iPAQ (download)
  - The iPAQ calls a Web service on the server with information about the records it is requesting. The server queries the database and sends a table to the iPAQ if records are available. The set of records is marked by a unique ID that is also sent to the iPAQ.
  - The iPAQ inserts the table into the local database on the device.
  - If the number of records inserted on the iPAQ database matches the number of records in the table from the server, then a download “receipt” is sent back to the server to acknowledge a successful transmission. The receipt contains the unique ID from the server that identifies the set of records.
  - Using the information in the receipt, the server marks the records as sent so they will not be transmitted again.
  - If the transmission fails, then the records will not be marked as sent on the server, and they will be sent again during the next transmission.
- Transmitting records from iPAQ to server (upload)
  - The iPAQ selects unposted records from the table.
  - The unposted records are sent to the server via a Web service.
    - When the server receives the table, the data are inserted into the database on the server.
    - The server responds to the iPAQ with the number of records added to the database to acknowledge the transmission.
  - If the response from the server indicates success (>0 records inserted in the database), the records on the iPAQ are marked as posted so that they will not be transmitted again.

- If the response from the server indicates a failure, the records on the iPAQ are not marked as posted, and they will be sent again during the next transmission.

Exhibit 2-1 provides a list of tables transmitted with a short description of the table and whether it is downloaded, uploaded, or both (see the section on Files and Structures for details on the definitions and contents of the tables).

Exhibit 2-1. Description of Transmitted Tables

Name	Description	Download	Upload
DU	Cases table	Yes	No
DUEvt	Case events table	Yes	Yes
Responses	Questionnaire responses table	No	Yes
Notes	Questionnaire and case notes table	No	Yes
AddressLog	Address change table	No	Yes
VerInfo	Verification information table (not used in GATS)	No	Yes
DelRecs	Deletion records table; indicates cases to be deleted from the iPAQ	Yes	No

Chapter 9 (section A–C) of the ***GATS Field Interviewer Manual*** provides a detailed description of the transmission process and screenshots. It describes the transmission over phone lines or wireless Internet and also discusses methods of data transfer using SD cards.

2.3 Updater Program

The Updater program facilitates updates of the applications on the iPAQ. When the Updater program is started, it checks the device for an existing connection. If there is no existing connection, the Updater will not operate. Currently, the Updater program allows updates for the following applications: CMS, GSS, and Xmit.

Updates are packaged as CAB files. CAB files on the iPAQ are equivalent to self-installing applications on a Windows system.

The functions of the Updater program are as follows:

- Check
  - When the user presses the **Check** button, the Updater application connects to the host servers and compares the currently installed version of the applications on the iPAQ to the version available on the server. If a new version is found, it is added to the list. After the program checks the application versions, the user can see the currently installed version and the available version by clicking on the application name in the list.

- Download
  - The user presses the **Download** button to begin downloading the updates. As mentioned earlier, the updates are packaged as CAB files. Each application that has a newer version available will have a single CAB file downloaded into an Updates folder (in a subfolder of the Updater application).
- Install
  - If updates were available and were downloaded to the device, the user can install the updates by selecting an item in the list and pressing **Install** (the **Install** button is disabled until an item is selected). This launches the CAB file for the selected item and installs the application via a wizard interface. After the installation of the CAB file, the item is removed from the list. If there are more updates, the process repeats until all updates are installed.
- Miscellaneous items
  - The Updater program is written in Visual Basic.NET using the Microsoft Compact Framework libraries.
  - The program uses the SQL Server CE database, a compact version of Microsoft SQL Server. The SQL CE database (SQL Server 2005 Compact Edition) is a shared database, with multiple data tables, that is used by all of the iPAQ software programs. It is stored as an encrypted database file.

Updater Instructions

The Updater program is used to update an iPAQ with the latest programs. Follow the steps below to update an iPAQ.

Before you start the update process, we recommend that you perform a soft reset on the iPAQ. This step will ensure no programs are running in the background that may interfere with the installation program.

To perform a soft reset:

- Locate the recessed **Reset** button on the bottom of the iPAQ Pocket PC.

Use the stylus to lightly press the **Reset** button.

Step 1: Using the transmit instructions, connect to the Central Office.

Step 2: While connected, return to the main screen.

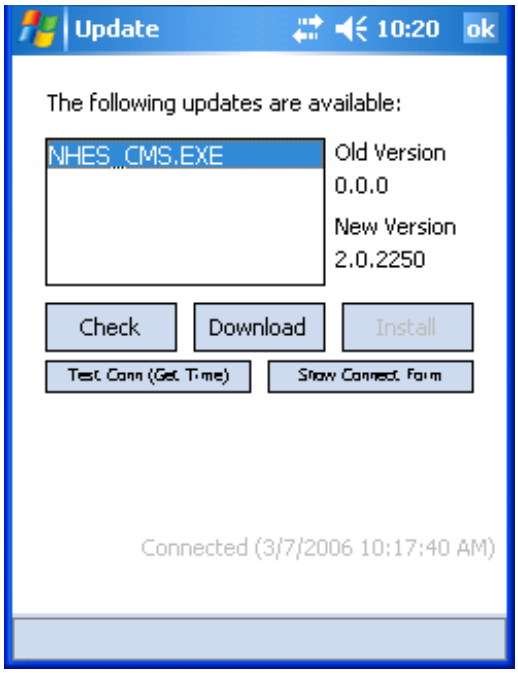
Step 3: Tap **Start** and display the list of programs on your iPAQ.

Step 4: Tap **Updater**.

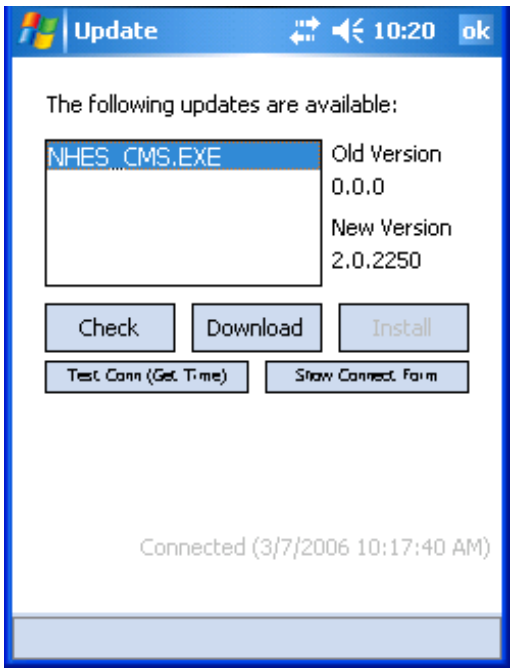
Step 5: Tap the **Check** box.

Step 6: If an updated file is available, the file name is displayed in the box. If there are no new updates, a pop-up window appears with this information.

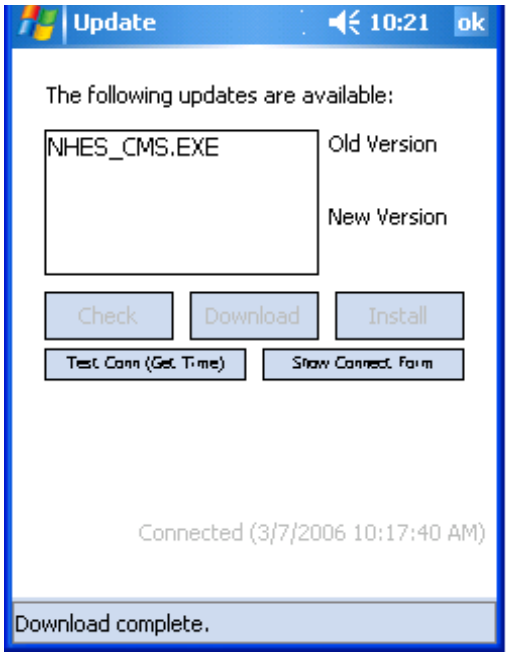
Step 7: Tap the **file name** in the box and highlight it.



Step 8: Tap **Download**.

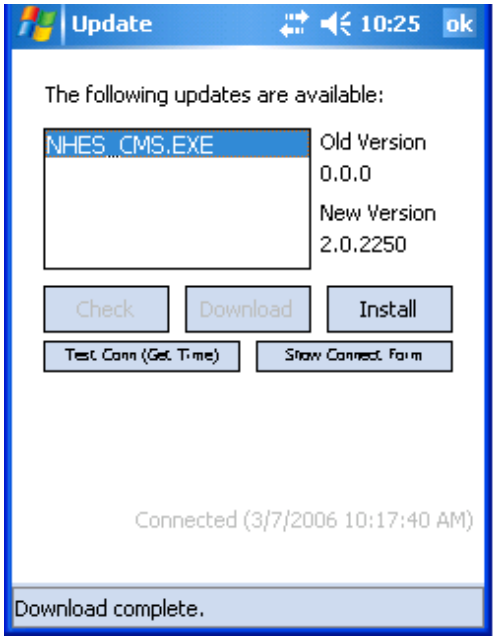


Step 9: The new file will be downloaded to the iPAQ. When the download is complete, the **Check**, **Download**, and **Install** buttons will be grayed out.

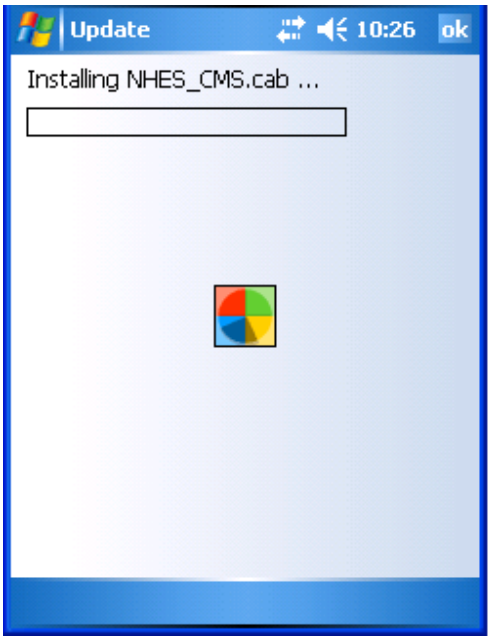


Step 10: Highlight the file name again (notice the **Install** button becomes active).

Tap **Install.**

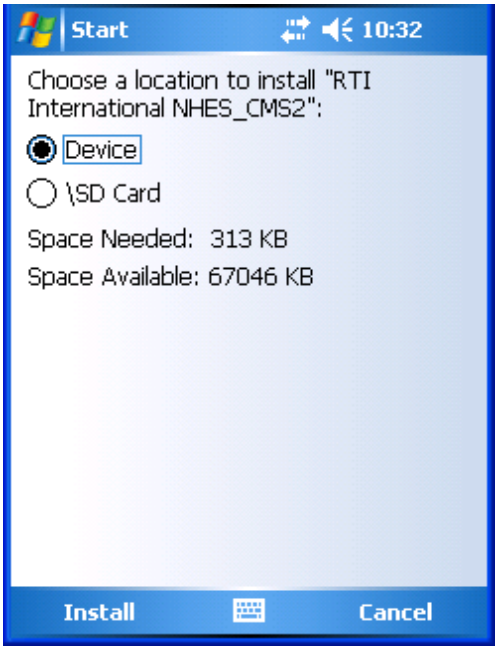


NOTE: The installer may display a message indicating that the previous version of the program will be removed before the new one is installed. If this message appears, click **OK** and continue.

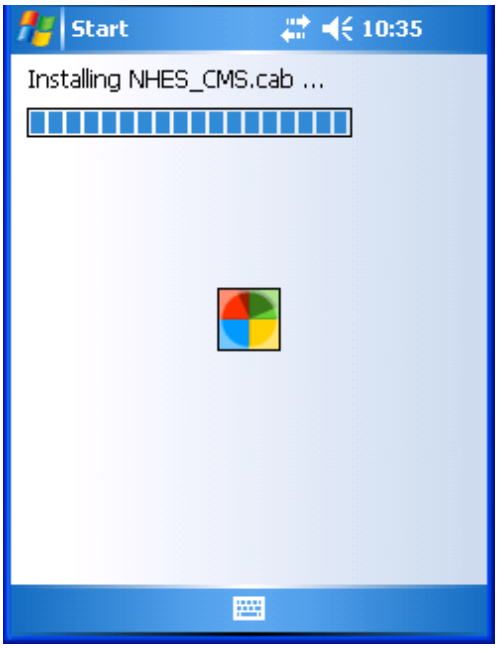


Step 11: This screen prompts you to save the file in the internal memory on the iPAQ or in the SD card. Make sure **Device** is selected.

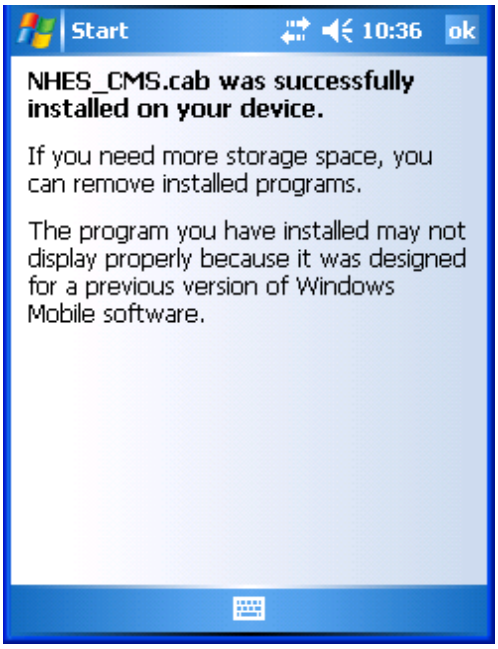
Tap **Install** in the bottom left corner.



Step 12: Wait while the program is installed.



Step 13: The screen indicates that the file was successfully installed. Tap **OK** in the top right corner to close the program.



Step 14: If there is more than one program to install, repeat steps 10 to 15 for each program. After all programs have been updated, the list will be empty. Close the Updater by clicking on the **OK** button in the top right corner.

Step 15: Finally, disconnect the iPAQ. Tap on the connection icon ( ) at the top of the screen to display the connection status. Tap the **Disconnect** button to close the connection.

3. Web Management Tools

Web management tools include case management, staff certification indicators, reports, and user maintenance. The Web-based case management allows Field Supervisors to track, distribute, and transfer cases to Field Interviewers. The Web tools can also be used to check and update certifications earned by Field Interviewers, view reports, and update user access to the system. The screen below shows the options available. This chapter explains the use of each option.

3.1 Case Management Options

Assign Cases. Field Supervisors use this screen to assign cases to a given Field Interviewer. Case assignment has two steps: assigning cases and releasing them to the iPAQ. Field Supervisors will use this screen for assigning basic cases.

Release Queued Cases. This screen is used to release cases to a Field Interviewer. Field Interviewers may have to pass project skills certification before they can receive cases on their iPAQ. The Web site will allow Field Supervisors to assign cases to Field Interviewers before the beginning of a period. However, the program checks whether the Field Interviewer has passed certification for that period before cases are released to him/her.

Recall Cases. This screen is used to recall a case from a Field Interviewer. To transfer a case from one Field Interviewer to another Field Interviewer, the Field Supervisor must use this screen to recall the case. After recalling a case, the Field Supervisor should contact the Field Interviewer, and the Field Interviewer should transmit. When the Field Interviewer transmits data, the transmit program first receives all the data for the recalled case and then removes the case from the iPAQ. When the recall is complete, a case is available for assignment to another Field Interviewer. Users are not allowed to recall completed cases. These cases appear on the grid with a red background, and there is no checkbox next to these cases to select.

Cancel Recall. This screen is used to cancel a recall operation. It is used when a Field Supervisor has accidentally issued a recall from the Web site. The Field Supervisor can use this option before the Field Interviewer transmits.

Pending Actions. This screen will show cases pending action at different stages. Field Supervisors can monitor cases with different pending actions, such as cases waiting in the queue for the Field Supervisors to release, cases waiting to be picked up by Field Interviewers, and cases set by the Field Supervisor for recall but that the Field Interviewer has not transmitted yet.

Unlock Codes. This screen generates unlock codes that are valid for a calendar date. They will not work for a different date. These codes are used on the iPAQ to unlock completed cases, purge old data, and so forth. When the Field Interviewer enters this code, a program running on the iPAQ will accept it.

3.2 Certification

A Field Interviewer may have to pass a project skills certification at the beginning of every period before working on cases. The Field Supervisor uses this screen to indicate whether a Field Interviewer has passed the certification. This screen is also used to indicate whether a Field Interviewer has successfully purged old data from previous periods before accepting new cases. A Field Interviewer cannot receive cases without passing these two checks. This option has not yet been used on GATS.

3.3 Reports

The following reports are available from the Web site:

- **Transmission Data**: this menu allows users to select reports on three data tables. Users can select case-level or Field Interviewer-level data:
  - event data
  - questionnaire data
  - transmission data
- **Basic Status Reporting**: this report is a summary report of the case status by interviewer. It shows percentage of cases completed, pending, and unworked.

  Users can develop reports tailored to their needs based on the SQL data that is available. The existing reports use MS ASP.net as a development platform.

3.4 User Maintenance

User maintenance includes the addition or removal of staff members from the project, changing the roles of staff on the project, and managing passwords. The following screens are used:

- Manage Users: this screen is used for maintaining roles for users.
- Change Password: this screen is used to change passwords to ensure data security.

4. General Survey System Engine

The General Survey System (GSS) is a suite of software tools designed to allow individuals to design applications to capture data using mobile computing devices. The GSS engine is a specific module that displays questions on the handheld and collects data. The GSS engine was designed to allow most, if not all, of the specifications for a data collection task to be done by simply describing the attributes of the data to be collected and specifying the flow of data collection steps in a single metadata repository (an MS Access database and its data tables). The sections that follow detail how you create the tables required for the GSS engine to generate a questionnaire and collect the data needed.

A data collection activity can be as simple as a Field Interviewer standing in a mall asking each passerby three or four questions—or it can be as involved as multihour face-to-face interviews in a person's home or office. It can also be an inventory task or check-in/check-out process. In GATS, the data collection activity is the collection of data required in two questionnaires: a household screener and an individual questionnaire. The information needed to define a data collection activity is:

- the exact specification of a set of questions to ask;
- the exact specification of the answers that are allowed;
- the exact specification of the sequence or order in which to ask the questions, given the state of the system at any given point in the data collection; and
- the instructions to the data collector or data provider.

The GSS engine considers a questionnaire to be a sequence of independent data collection screens. The screen is the building block that the system uses to create questionnaires. A screen is the display that the Field Interviewer sees. It can contain instructions, text to read to the respondent, a field or fields that call for data input, navigation controls, menus, and status information. A questionnaire is then defined by tying together a sequence of screens to collect the information desired. In most cases, a screen collects one piece of data, for example, a respondent's first name, age, or response to a question. The GSS engine treats all data collected from a single screen as a unit. The data from a screen are saved as a row in a permanent data table on the Pocket PC and serve as the source to build analysis files of the collected data.

4.1 Overall Design

The general design of the GSS engine is based on the following key concepts:

- The unit of data collection is the screen.
- A screen has the following basic components:
  - a field for text (question or instructions),
  - one or more fields for data input (may contain lists to select from or blank fields to enter text or numbers), and
  - a set of menu items to control navigation among the screens:

    A **Next** button allows users to go forward one screen.

    A **Back** button allows users to go back one screen.

A **Language** button allows users to toggle between two languages.

A utility menu that includes various options (e.g., Break Off, Case Info, enter a note).

A **Help** button (QxQ) that can provide help information about the question on a given screen.

- All the data items/fields collected on a single screen are stored in a single character variable. Multiple items are separated and terminated by the | character (e.g., field1|field2|field3|).
- Each screen generates one and only one data row in an output data table of response data each time the user exits the screen by tapping the **Next** button.
- The data row captures the data item(s) from the screen plus control data (time stamp, language used, validity flag, etc.) each time the user exits the screen by tapping the **Next** button.
- Some special screens do not generate any output data and/or may not be visible to the user (screens that invoke internal computations or control looping).

A questionnaire is developed by creating a collection of these screens, providing the metadata for each screen, and defining the sequence they are to be shown to the data collector or data provider.

The iPAQ screenshot shown here (screen A02A) displays a sample GSS engine *list* screen type. In a *list* screen, the Field Interviewer reads the question text (for this example, “What is the month of your date of birth?”) and then, depending upon the answer received from the respondent, the Field Interviewer selects (taps) one of the answer options (01, 02, etc.). The sliders at the side and bottom of the screen indicate that the Field Interviewer can scroll up and down or left and right to see more of the question or answer choices, if needed. The Field Interviewer then simply taps the answer option the respondent selected and then taps **Next** to go on to the next screen. The elements of the screen are described in **Exhibit 4-1**.

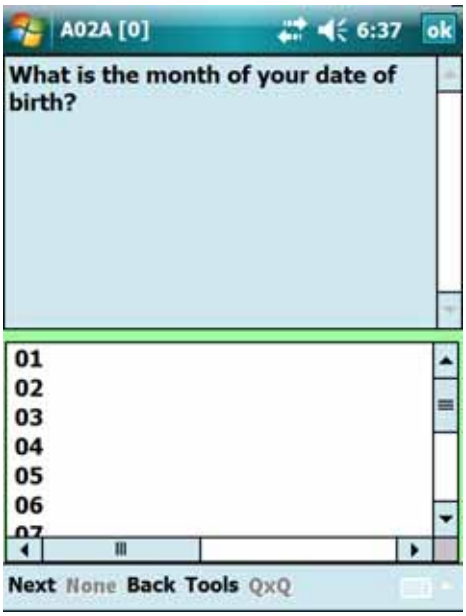


Exhibit 4-1. GSS Engine Screen Elements

Item	Element Name	Value Shown on Screen A02A
Screen name	QID	A02A
Nesting or loop level	Instance	[0]
Question text	Qtext	What is the month…
Answer text choices	Atext(i)	01, 02, 03, 04, 05, 06, 07,…
Navigation buttons		
Move to next screen	Next	Next
Move back one screen	Back	Back
Language toggle	LangToggle	None
Utility functions	ToolsMenu	Tools
A help button	QxQ	QxQ

The GSS engine supports a number of screen types that the questionnaire developer can use to build the data collection application. **Exhibit 4-2** lists the screen types that are supported. They fall into three categories:

- Data capture: List, All That Apply (ATA), Numeric (Num), Text, Date, Phone # (Phn#), Name, Inventory (Invent), SSN, Feet and Inches (Feet), Address (Addr), Address1 (Addr1)
- Data display: Info
- Navigation or computation: Start, End, Loop Start (Loop), Loop End (Lend), Compute (Comp)

4.2 Example 1—Program a Small Questionnaire

An example of a small questionnaire may help to show how a developer might build a data collection questionnaire or form. This example is based on a few simple questions from the GATS *Individual Questionnaire*. The paper specifications for these questions are shown in **Exhibit 4-3**.

Exhibit 4-2. GSS Engine Screen Types

Screen Type ID	Screen Description	Generates Data Row	# of Input Fields	Type Data Collected
List	Select one item from a fixed list of choices	Yes	1	Code associated with the selected choice
ATA	Select one or more items from a fixed list of choices	Yes	Up to 31	Single numeric code based on which mix of items selected 0 to 2^31
Num	Enter a single numeric item	Yes	1	Numeric; can be floating point or restricted to integer
Text	Enter a single text item	Yes	1	Character string, maximum length 32,767; only first 255 will be stored
Date	Enter a single date item containing month, day, year	Yes	1	Date stored as character string in mm/dd/yyyy format
Phn#	Collects a U.S. phone number	Yes	1	Standard U.S. 10-digit phone number including area code
Name	Collects a full name	Yes	2	First name and last name
Invent	Collects 4 text data items	Yes	4	Designed for generic multi-item screen, all fields as text, no range or format, user can specify labels
Info	Presents only text information to user; does not collect anything	Yes	0	No data collected on this screen
SSN	Collects a U.S. Social Security number	Yes	1	Collects a standard U.S. Social Security number xxx-yy-zzzz
Feet	Collects 2 fields (feet and inches)	Yes	2	Treats each field as numeric, requires inches field to be >=0 and <12
Start	Required screen—must be the first screen executed	No	0	No data collected
End	Required screen—must be the last screen executed	No	0	No data collected
Addr	Collects address data	Yes	8 items	Collects city, state, zip code, street #, RR #, PO Box #, apt. #, street name
Addr1	Collects address data	Yes	5 items	Collects 2 lines of street info and city, state, and zip
Loop	Not visible to user	No	0	Used to define a loop start
Lend	Not visible to user	No	0	Used to define a loop end
Comp	Not visible to user	No	0	Used to implement user logic calls

Exhibit 4-3. GATS Individual Questionnaire Subset

Individual Questionnaire Subset
INTRO1 You have been selected to participate in the Global Adult Tobacco Survey sponsored by [FILL COUNTRY SPONSORING AGENCY]. I would like to start now. Is that okay?
INTRO2 CONSENT HAS BEEN READ TO PARTICIPANT. YES ..... ☐ 1 NO ..... ☐ 2 [GO TO INTRO1]
INTRO3 CONSENT HAS BEEN OBTAINED [VERBAL] YES ..... ☐ 1 NO ..... ☐ 2 [END INTERVIEW]
A00 I am going to first ask you a few questions about your background.
A01 RECORD GENDER FROM OBSERVATION. ASK IF NECESSARY. MALE..... ☐ 1 FEMALE ..... ☐ 2
A02a What is the month of your date of birth? 01 ..... ☐ 1 02 ..... ☐ 2 03 ..... ☐ 3 04 ..... ☐ 4 05 ..... ☐ 5 06 ..... ☐ 6 07 ..... ☐ 7 08 ..... ☐ 8 09 ..... ☐ 9 10 ..... ☐ 10 11 ..... ☐ 11 12 ..... ☐ 12 DON'T KNOW .... ☐ 77 REFUSED ..... ☐ 99

(continued)

Exhibit 4-3. GATS Individual Questionnaire Subset (continued)

A02b What is the year of your date of birth? IF DON'T KNOW, ENTER 7777 IF REFUSED, ENTER 9999 [VALIDATION: CALCULATED DATE OF BIRTH >= 15 YEARS OLD] [IF MONTH OR YEAR = DON'T KNOW OR REFUSED, GO TO A03. OTHERWISE GO TO A04.]
A03 How old are you? IF RESPONDENT IS UNSURE, PROBE FOR AN ESTIMATE AND RECORD AN ANSWER. IF REFUSED, ENTER 999 _____ [RANGE: 15 – 125] [VALIDATION: A03 >= 15] [IF A03=999, GO TO A04]
A03a WAS RESPONSE ESTIMATED? YES ..... ☐ 1 NO ..... ☐ 2 DON'T KNOW .... ☐ 7
A04 What is the highest level of education you have completed? SELECT ONLY ONE CATEGORY [ADJUST CATEGORIES FOR SPECIFIC COUNTRY] NO FORMAL SCHOOLING..... ☐ 1 LESS THAN PRIMARY SCHOOL COMPLETED..... ☐ 2 PRIMARY SCHOOL COMPLETED..... ☐ 3 LESS THAN SECONDARY SCHOOL COMPLETED ..... ☐ 4 SECONDARY SCHOOL COMPLETED ..... ☐ 5 HIGH SCHOOL COMPLETED ..... ☐ 6 COLLEGE/UNIVERSITY COMPLETED..... ☐ 7 POST GRADUATE DEGREE COMPLETED ..... ☐ 8 DON'T KNOW ..... ☐ 77 REFUSED ..... ☐ 99
I00 Those are all of the questions I have. Thank you very much for participating in this important survey.

To develop this paper form as a GSS application, you can use 10 screens. The screen names, order of presentation, and screen types are shown below:

Question Name	Screen Type
1. Intro1	Info
2. Intro2	List
3. Intro3	List
4. A00	Info
5. A01	List
6. A02b	Num
7. A03	Num
8. A03a	List
9. A04	List
10. I00	Info

No loops or routing paths are used in this very simple example. Three different GSS screen types could be used for the application (Info, List, and Num). To build the screens needed for this example, place the information in Access tables. **Exhibits 4-4**, **4-4a**, **4-5**, and **4-6** show the four Access tables (Questions, Texts, Answers, and Initialization Data-INI, respectively) required to build this example. Of these tables, the INI table is a repository for general parameters for the whole questionnaire and GSS setup for its execution. The other three tables contain the question-level information that defines the questionnaire. The Questions table contains the information that pertains to the question and it has links to the two other tables for additional information. The Texts table is linked by the variable QID to the Questions table and contains the text for a given question. In the Texts table you must provide one row for each QID for each language you wish to support. If you plan to use English, Spanish, and French, then every QID with text needs three rows. The Answers table is somewhat similar in that it contains the information that defines the List and All that Apply answer choice lists or Answer Sets. Again, these lists can be in as many languages as needed. Each Answer Set is repeated for every language required. In **Exhibit 4-4** (the Questions table), each row represents the metadata for a given question/screen. Every GSS questionnaire must start with a Start screen (Qsequence 1 in **Exhibit 4-4**) and end with an End screen (Qsequence 25000 in **Exhibit 4-4**). The first two columns of the Questions table provide ID information for the system, a sequence number that provides the default order for listing the questions, and a name for the screen (QID).

Exhibit 4-4. Screen Information: Questions Table

QSequence	Qid	QNext	Qlogic	QType	Special Instructions	Qanswer	RangeLo	RangeHi
1	Start	Intro01		COMP		None	0	0
10	Intro01	Intro02		INFO		None	0	0
20	Intro02	Intro03	If {Intro02}="2" then goto Intro01;	LIST		YESNO	0	0
30	Intro03	A00	If {Intro03}="2" then goto End;	LIST		YESNO	0	0
50	A00	A01		INFO		None	0	0
60	A01	A02a		LIST		SEX	0	0
80	A02a	A02b		LIST		MONTHLIST	0	0
90	A02b	A04	If {A02a}="77" or {A02b}="7777" or {A02a}="99" or {A02b}="9999" then goto A03;	NUM	NoDKRE,Integer, RngInclude=7777; 9999;	None	1900	2000
100	A03	A03a	If {A03}="999" then goto A04;	NUM	NoDKRE,Integer, RngInclude=999;	None	15	125
120	A03a	A04		LIST		YESNODK	0	0
130	A04	I00		LIST	FontSize=7	EDUCATION	0	0
160	I00	End		INFO		None	0	0
25000	End	End		END		None	0	0

Exhibit 4-4a. Question Text: The Texts Table

Pkey	Qid	Language	QText
1	Start	English	Ver.: 2009 06 26 005
2	Intro01	English	You have been selected to participate in the Global Adult Tobacco Survey sponsored by [FILL COUNTRY SPONSORING AGENCY]. I would like to start now, is that okay?
3	Intro02	English	CONSENT HAS BEEN READ TO PARTICIPANT.
4	Intro03	English	CONSENT HAS BEEN OBTAINED [VERBAL].
5	A00	English	I am going to first ask you a few questions about your background.
6	A01	English	RECORD GENDER FROM OBSERVATION. ASK IF NECESSARY.
7	A02a	English	What is the month of your date of birth?
8	A02b	English	What is the year of your date of birth? IF DON'T KNOW, ENTER 7777. IF REFUSED, ENTER 9999.
9	A03	English	How old are you? IF RESPONDENT IS UNSURE, PROBE FOR AN ESTIMATE AND RECORD AN ANSWER .IF REFUSED, ENTER 999
10	A03a	English	WAS RESPONSE ESTIMATED?
11	A04	English	What is the highest level of education you have completed? SELECT ONLY ONE CATEGORY [ADJUST CATEGORIES FOR SPECIFIC COUNTRY].
12	I00	English	Those are all of the questions I have. Thank you very much for participating in this important survey.
13	End	English	END OF SURVEY [TAP EXIT TO CLOSE SURVEY OR TAP BACK TO GO BACK].

Exhibit 4-5. Answer Text and Codes: Answers Table

UID	AID	Language	ASequence	ACode	AText
558	EDUCATION	English	1	1	NO FORMAL SCHOOLING
559	EDUCATION	English	2	2	LESS THAN PRIMARY SCHOOL COMPLETED
560	EDUCATION	English	3	3	PRIMARY SCHOOL COMPLETED
561	EDUCATION	English	4	4	LESS THAN SECONDARY SCHOOL COMPLETED
562	EDUCATION	English	5	5	SECONDARY SCHOOL COMPLETED
563	EDUCATION	English	6	6	HIGH SCHOOL COMPLETED
564	EDUCATION	English	7	7	COLLEGE/UNIVERSITY COMPLETED
565	EDUCATION	English	8	8	POST GRADUATE DEGREE COMPLETED
566	EDUCATION	English	9	77	DON’T KNOW
567	EDUCATION	English	10	99	REFUSED
568	YESNODK	English	1	1	YES
569	YESNODK	English	2	2	NO
570	YESNODK	English	3	7	DON'T KNOW
571	MONTHLIST	English	1	1	01
572	MONTHLIST	English	2	2	02
573	MONTHLIST	English	3	3	03
574	MONTHLIST	English	4	4	04
575	MONTHLIST	English	5	5	05
576	MONTHLIST	English	6	6	06
577	MONTHLIST	English	7	7	07
578	MONTHLIST	English	8	8	08
579	MONTHLIST	English	9	9	09
580	MONTHLIST	English	10	10	10
581	MONTHLIST	English	11	11	11
582	MONTHLIST	English	12	12	12
583	MONTHLIST	English	13	77	DON'T KNOW
584	MONTHLIST	English	14	99	REFUSED
585	SEX	English	1	1	Male
586	SEX	English	2	2	Female
589	YESNO	English	1	1	YES
590	YESNO	English	2	2	NO

Exhibit 4-6. Initialization Data: INI Table for Survey Form

Variable Name	Variable Value	Comment
Breakoff	Yes	Flag to control Breakoff option (Yes/No)
DisplayOldData	Yes	If old answers exist, show them on backup (Yes/No)
FastForwardOK	No	Toggle to turn on or off Fast Forward option: Yes or No
Lang0Font	Tahoma	Font to be used when using Language 0
Lang0label	Eng	Language 0 menu button label
NumberofSurveys	1	Number of different surveys in this project (1 to n)
NumLanguages	1	Number of languages used
Qversion	Version 1.0 March. 2008	Version of the INI file
QxQOption	Yes	Turn on QxQ option (Yes/No)
RosterLabel1	Name|Age|B Mo|B Yr|Gender|Smkr|	
StartUpLanguage	0	Language to use of Q startup, 0=English, 1=Other
SurveyID	GATS_Example	Unique ID for this survey
TextToCaps	Yes	Convert all text entry to CAPS (Yes/No)

The next column in the Questions table, Qnext, defines the question to go to next; that is, after you press the next menu item on the given screen. In the Qlogic column, you can enter GSS programming commands in a Visual Basic-like programming language. These logic statements allow you to create variables, alter program flow, and output data. **Appendix A** discusses the Qlogic programming language and how to code commands for this function in GSS.

In column Qtype, specify the type of screen. The special instructions column allows you to provide some formatting directives to the system (see Qsequence 90 for an example and **Exhibit 4-7** for a complete list of the available formatting options). In column Qanswer, specify the name of the answer set (see Answers table) for List or ATA type screens. This is the name of a set of answers to be used as answer options. There must be a set (one or more) of these answers in the Answers table to match the reference in the Questions table. The next two columns, RangeLo and RangeHi, provide a place to specify a minimum and maximum range for numeric type screens (see row 90 for an example). If the answer to a question falls outside the range specification (RangeLo ≤ Answer ≤ RangeHi), it will generate validation errors and force you to input data within the range. It is also possible to specify ranges as a list. **Exhibit 4-8** details the functions and requirements for each column of a Questions table. The question text is stored in a separate table, the Texts table (see **Exhibit 4-4a**). The Texts table has a row for every QID and language. Since this example is English only, the Texts table has a row for each QID in the Questions table that has question text.

Exhibit 4-7. GSS Engine Special Instructions Options

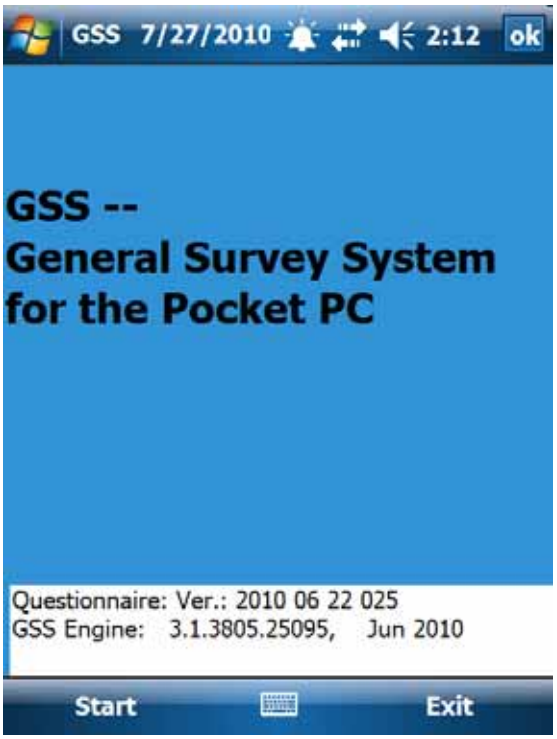
Option Name	Function	Applicable Screen Types	Parameters
Integer	Forces input to whole numbers only (no decimal points)	Num	None
NODK	Disables use of Don't Know codes (?)	Text, Num	None
NORE	Disables use of Refusal option (!)	Text, Num	None
NODKRE	Disables both Don't Know (?) and Refusal codes (!)	Text, Num	None
N/A=Yes	Allows use of "*" symbol to indicate not applicable	Text, Num, SSN, Phn	None
Fontsize=xx	Allows user to specify font size for answer text in list screens	List, ATA	8 to 20
Textsize=xx	Allows user to specify font size for text field	All screens with a text field	8 to 20
SIPtype=xxxx	Allows user to select kind of popup or soft keyboard shown	Text, Num, SSN, PHN	Char or Num
ATAf=x	Number of non-answer (Refused, Don't Know, etc.) codes in ATA list	ATA	Integer 1 to 31
RngInclude	Directive to include values into a range check on a numeric variable in addition to that specified by RangeHi and RangeLo	Num	E.g., RngInclude=99; 98; 97, this will add the values 99, 98, and 97 to the valid range
CAPS	Directive to capitalize all text entered	Text, Name, Addr	
RngExcp	Directive to exclude values from a range check on a numeric variable	Num	E.g., RngExcp=2; 8, excludes 2 and 8 as legal values in a specified range
LoadCase	Special code to preload Case info from CMS	ADDR	None
NoBackup	Restricts backup at this screen, does not allow Field Interviewer to backup over the screen	All	None
AtextFSize	Allows setting the answer text font size for both language 0 and language 1	List, ATA	Fontsize lang 0, fontsize lang 1, e.g., ATextFSize=(08,10);
QtextFSize	Allows setting the question text font size for both language 0 and language 1	List, ATA	Fontsize lang 0, fontsize lang 1, e.g., ATextFSize=(08,10);
NoBreakOff	Restricts Break Off at this screen, does not allow Field Interviewer to use the Break Off menu option on this scfreen	All	None

Exhibit 4-8. GSS Screen Properties

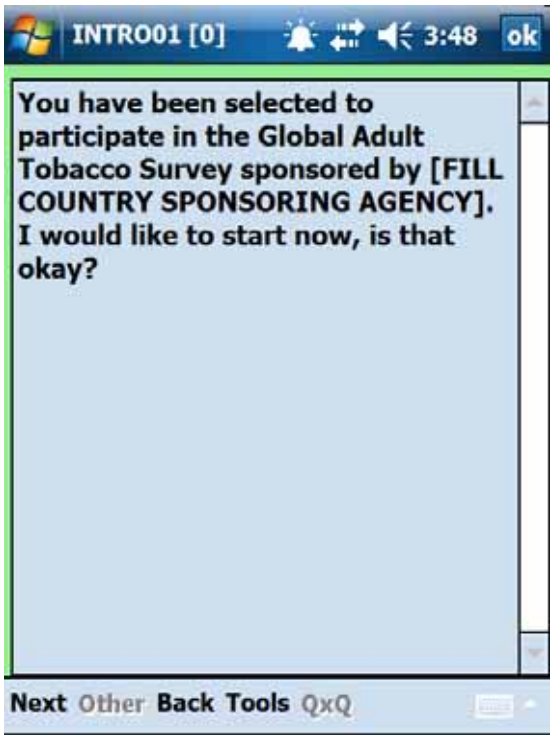
Attribute Name	Function	Requirements	Notes
Sequence #	Controls the sort order of screens; not used for flow, just presentation of rows for viewing or printing	Number in the range 1.000 to 99999.999	
QID	Name of screen	Must be unique within questionnaire and <= 20 characters	
Qtype	Type of screen	See **Exhibit 4-2** for list of available types	
AnswerSet	Name of list of an answer set of response options for List or ATA type questions (<= 20 characters)	Verbatim text that will be displayed in List and ATA type screens	Can use the same answer set for multiple questions
Logic	GSS programming language for logic and flow control	See **Appendix A** for discussion of coding conventions for Qlogic	
Special Instructions	List for formatting or other special options that apply to the given screen (see **Exhibit 4-7**)		
RangeLo	Range for Qtype Num used for a validity check	Numeric value	Low value can be any real number
RangeHi	Range for Qtype Num used for a validity check	Numeric value	High value can be a real number >= RangeLo
Loop Info	Parameters used to define DO loops		
LoopStart	Symbol or constant that defines starting loop counter value		
LoopEnd	Symbol or constant that defines ending loop counter value		
LoopGoto	Screen to go to when loop first starts or restarts the loop		
Roster #	Roster # to store a copy of this answer	1, 2, 3, or 4	Typically used to build grids of data collected in loops for later editing or display
Roster Col	Column # of the roster to store a copy of the data	0 to 20	The row # is determined by the instance level of the questionnaire but user may choose column

The set of specifications provided previously leads to the following set of GSS screens. Note that the first and last screens come from the Start and End screens defined in the Questions table.

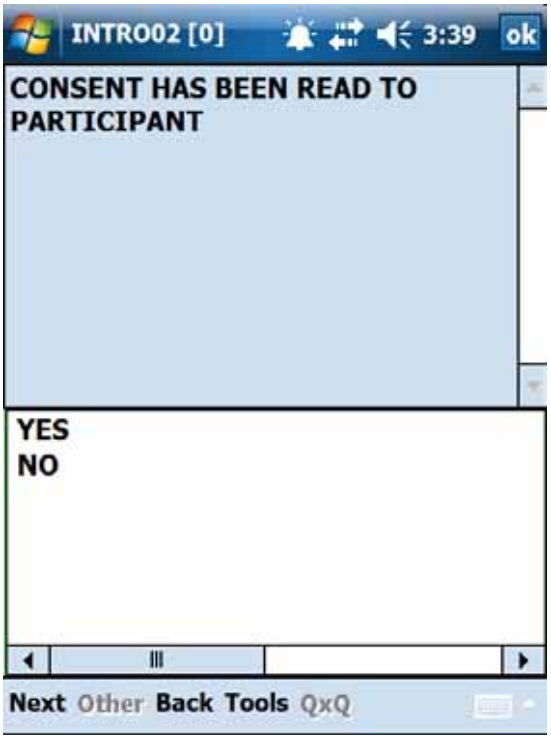
GSS Start Screen



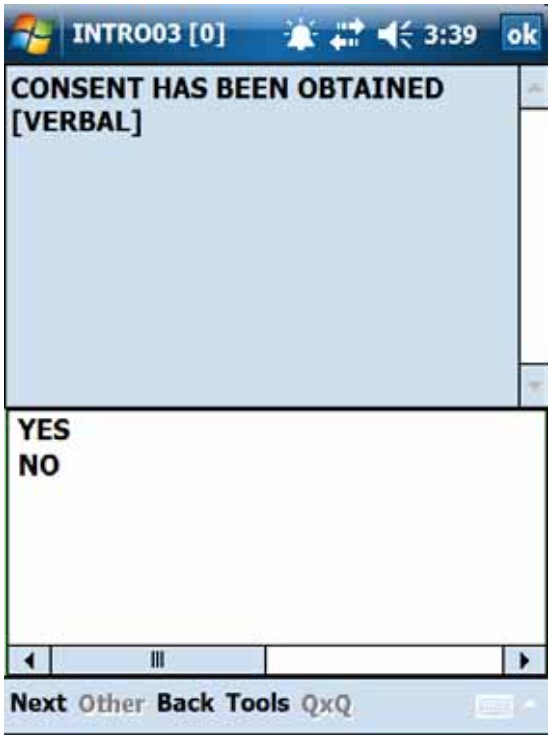
Intro01 Screen



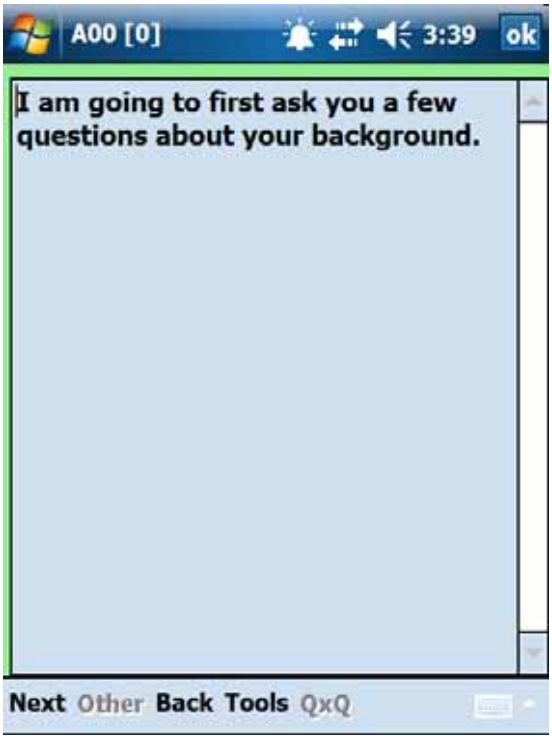
Intro02 Screen



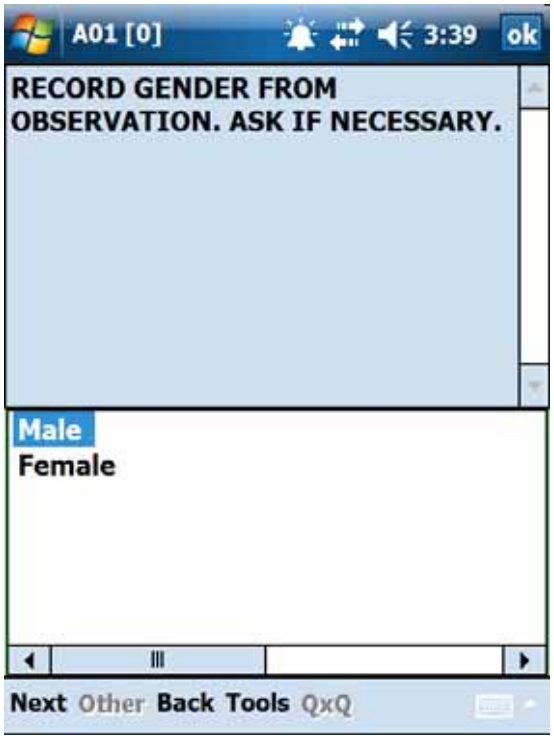
Intro03 Screen



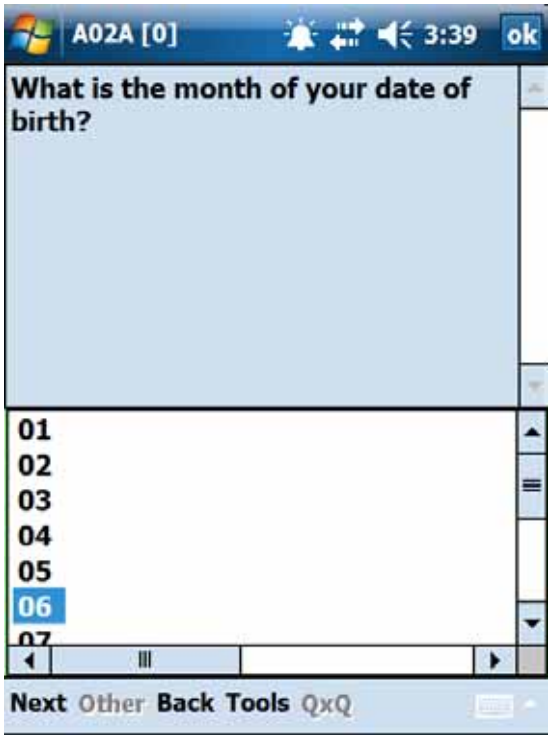
A00 Screen



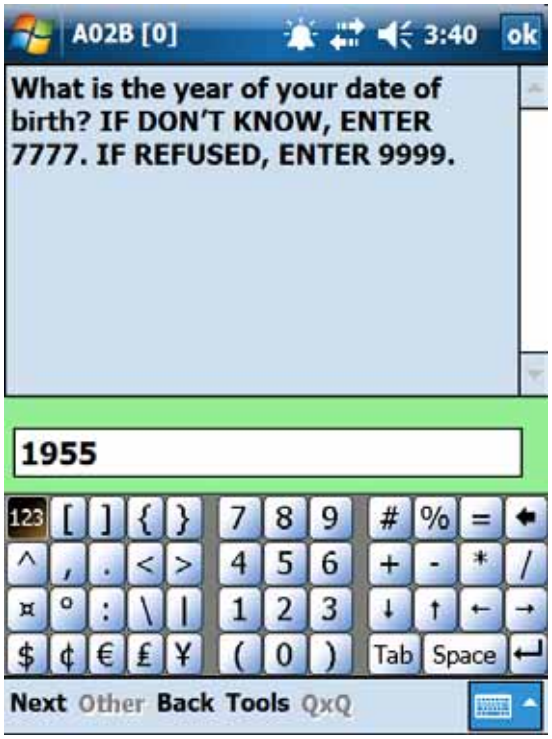
A01 Screen



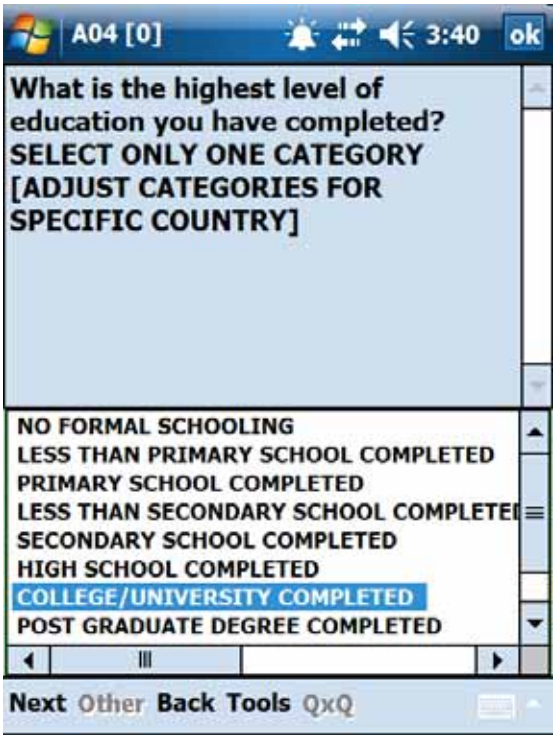
A02A Screen



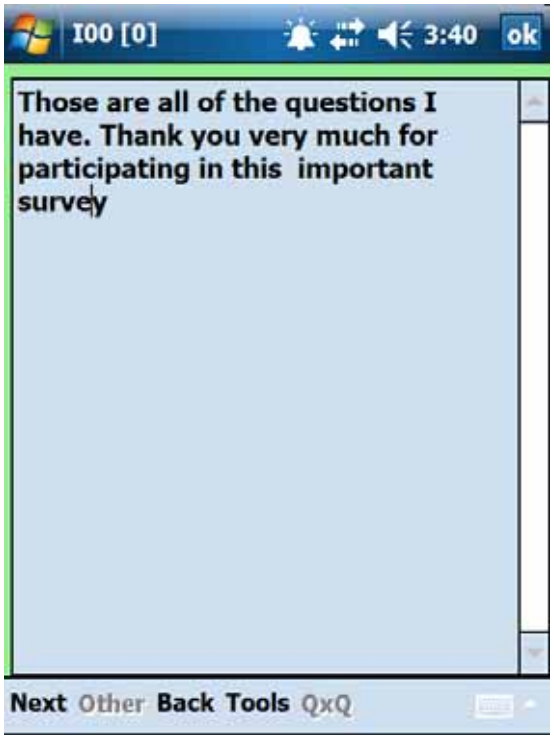
A02B Screen



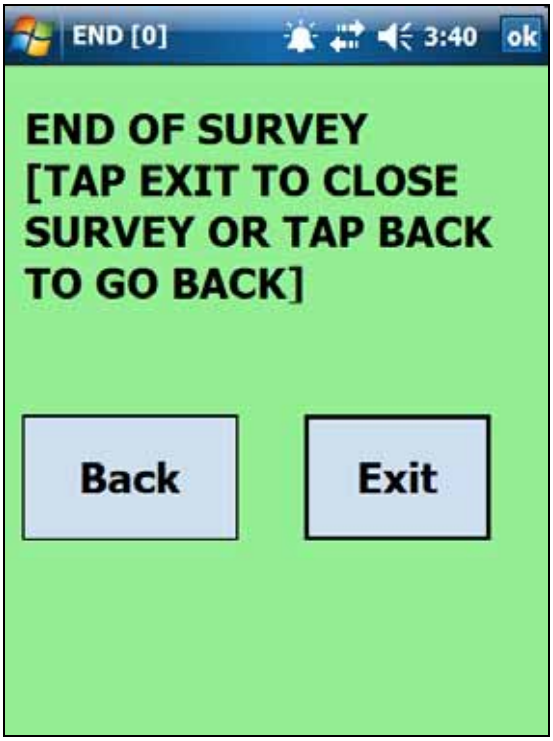
A04 Screen



I00 Screen



End Screen



4.3 Example 2—Using GSS to Program the GATS Household Questionnaire

A second, more complicated example provides more details on GSS programming. This example is based on the screens required to implement the GATS *Household Questionnaire*. The GATS *Household Questionnaire* requires collection of data on the members of a household and selection of one of those members to participate in the individual interview. The paper version of this form is shown in **Exhibit 4-9**.

Exhibit 4-9. Household Questionnaire

INTRO

[THE HOUSEHOLD SCREENING RESPONDENT MUST BE 18 YEARS OF AGE OR OLDER AND YOU MUST BE CONFIDENT THAT THIS PERSON CAN PROVIDE ACCURATE INFORMATION ABOUT ALL MEMBERS OF THE HOUSEHOLD.

IF NEEDED, VERIFY THE AGE OF THE HOUSEHOLD SCREENING RESPONDENT TO MAKE SURE HE/SHE IS 18 YEARS OF AGE OR OLDER.]

INTRO1

An important survey of adult tobacco use behavior is being conducted by the (FILL COUNTRY SPONSORING AGENCY) throughout (FILL COUNTRY NAME) and your household has been selected to participate.

All houses selected were chosen from a scientific sample and it is very important to the success of this project that each participates in the survey. All information gathered will be kept strictly confidential. I have a few questions to find out who in your household is eligible to participate.

HH1

First, I'd like to ask you a few questions about your household. In total, how many persons live in this household?

[INCLUDE ANYONE WHO CONSIDERS THIS HOUSEHOLD THEIR PRIMARY PLACE OF RESIDENCE.]

_____ [RANGE: 0 – 50]

[IF HH1 = 00, GO TO NOELIGIBLE]

HH2

How many of these household members are 15 years of age or older?

_____ [RANGE: 0 – 20]

[VALIDATION: HH2 <= HH1 (IF NOT, GO TO TooMany)]
[IF HH2 = 00, GO TO NOELIGIBLE]
[IF HHType = MALE OR FEMALE, GO TO HH3. IF HHType = BOTH, GO TO HH4both]

TooMany

[YOU CAN'T/SHOULDN'T HAVE MORE PEOPLE >= 15 YEARS OLD THAN THERE ARE TOTAL HH MEMBERS; PLEASE DOUBLE CHECK THE ANSWERS SO FAR.]

[GO TO HH2]

(continued)

Exhibit 4-9. Household Questionnaire (Page 2 of 5)

HH3

How many {FILL HHType: male/female} household members are 15 years of age or older?

_____ [RANGE: 0 – 20]

[VALIDATION: HH3 <= HH2 (IF NOT, GO TO TooMany2)]
[IF HH3 = 00, GO TO NOELIGIBLE]
[ELSE, GO TO HH4]

TooMany2

[YOU CAN'T/SHOULDN'T HAVE MORE {HHtype}s >= 15 THAN THERE ARE TOTAL HH MEMBERS >= 15. PLEASE DOUBLE CHECK THE ANSWERS SO FAR.]

[GO TO HH3]

HH4
I now would like to collect information about only the {FILL HH3} {FILL HHType: male(s)/female(s)} that live in this household who are 15 years of age or older.

Let's start listing the {FILL HHType: male(s)/female(s)} from oldest to youngest.

[GO TO HH4a]

HH4both

I now would like to collect information about only these persons that live in this household who are 15 years of age or older.

Let's start listing them from oldest to youngest.

HH4a
What is the {FILL: oldest/next oldest} person's first name?

_____

HH4b
What is this person's age?

[IF RESPONDENT DOESN'T KNOW, PROBE FOR AN ESTIMATE]

_____ [RANGE: 15 – 110]
[IF HH4b >= 15 and <= 17, GO TO HH4c. OTHERWISE, GO TO HH4d]

(continued)

Exhibit 4-9. Household Questionnaire (Page 3 of 5)

HH4c

What is the month of this person's date of birth?

01 ..... ☐ 1
02 ..... ☐ 2
03 ..... ☐ 3
04 ..... ☐ 4
05 ..... ☐ 5
06 ..... ☐ 6
07 ..... ☐ 7
08 ..... ☐ 8
09 ..... ☐ 9
10 ..... ☐ 10
11 ..... ☐ 11
12 ..... ☐ 12
DON'T KNOW ..... ☐ 77
REFUSED ..... ☐ 99

HH4cYEAR

What is the year of this person's date of birth?

[IF DON'T KNOW, ENTER 7777
IF REFUSED, ENTER 9999]

_____ [RANGE: 1900 – 2000]

[IF HH4c = 77 OR HH4c = 99 OR HH4cYear = 7777 OR HH4cYear = 9999, THEN GO TO HH4d]
[VALIDATION: CALCULATED DATE OF BIRTH >= 15 YEARS OLD (IF NOT, GO TO ValidateAge)]
[ELSE, GO TO HH4d]

ValidateAge

[AGE CALCULATED FROM BIRTH MONTH AND BIRTH YEAR IS LESS THAN 15. PLEASE DOUBLE CHECK THESE ANSWERS.]

[GO TO HH4c]

HH4d

{IF HHType = BOTH: Is this person male or female?}

{IF HHType = MALE OR FEMALE: [RECORD GENDER (FOR VERIFICATION IF NECESSARY)]}

MALE ..... ☐ 1
FEMALE..... ☐ 2

[IF HHType = BOTH, GO TO HH4e]
[IF HHType = MALE OR FEMALE, DO VALIDATION: HH4d = HHType (IF NOT, GO TO WrongGender)]

(continued)

Exhibit 4-9. Household Questionnaire (Page 4 of 5)

WrongGender

[YOU SHOULD BE ROSTERING ONLY {HHType_0}s; PLEASE DOUBLE CHECK THIS DATA.]

[GO TO HH4d]

HH4e

Does this person currently smoke tobacco, including (FILL APPROPRIATE COUNTRY EXAMPLES: cigarettes, cigars, pipes)?

YES..... ☐ 1
NO..... ☐ 2
DON'T KNOW..... ☐ 7
REFUSED..... ☐ 9

EDITROSTERINTRO

[IF YOU NEED TO REVIEW THE ROSTER, SELECT "ROSTER" FROM THE TOOLS MENU.

TAP THE BACK BUTTON IF YOU NEED TO MAKE CHANGES.

TAP THE NEXT BUTTON TO SELECT THE RESPONDENT.]

HH5

[NAME OF THE SELECTED ELIGIBLE PERSON IS:

{FILL SELECTED HH MEMBER'S FIRST NAME}

ASK IF THE SELECTED RESPONDENT IS AVAILABLE AND IF SO, PROCEED TO THE INDIVIDUAL QUESTIONNAIRE.

IF THE SELECTED RESPONDENT IS NOT AVAILABLE, MAKE AN APPOINTMENT AND RECORD IT AS A COMMENT ON RECORD OF CALLS.]

[SET {HH5Flag} = "1"]
[GO TO CodeEvents]

NOELIGIBLE

[THERE ARE NO ELIGIBLE HOUSEHOLD MEMBERS.

THANK THE RESPONDENT FOR HIS/HER TIME.

THIS WILL BE RECORDED IN THE RECORD OF CALLS AS A CODE 201.]

[Set {NoEFlag} = "1";]

CodeEvents

(continued)

Exhibit 4-9. Household Questionnaire (Page 5 of 5)

```
if {HH5Flag} = "1" then set {EventCode} = "200";
if {NoEflag} = "1" then set {EventCode} = "201";
if {HH5Flag} = "1" then set {EventComment} = "Screener Complete";
if {NoEflag} = "1" then set {EventComment} = "Screener Complete No Eligibles";
```

Exhibits 4-10 and **4-11** show the complete GSS programming specifications for the handheld implementation of this paper form. As shown previously, these specifications are stored in Microsoft Access tables (Questions, Answers, Texts). The screen types used are Comp, Info, Num, List, Text, Loop, and Loop End. The visible screens generated by these specifications and a sample set of specific answers provided are shown as screenshots following **Exhibit 4-11**. (Comp screens do not display any information to the Field Interviewer and, depending upon how questions are answered, many paths are possible through the questionnaire and this is only one possible path.) Specific lines of the Questions table (**Exhibit 4-10**) reinforce features introduced before and illustrate several new features. Specific details to note are as follows:

- The Q sequence numbers are not sequential or in any specific order, but they are unique. They serve only to order the display of the rows in the Questions table.
- The Instance Level is shown on the top line of the screen after the QID in square brackets. It is 0 when not in a loop and >0 when inside a loop. It can be seen to be changing from 0 in the loop over members of the household (see screenshots HH4A–HH4E below).
- The QID names are ≤20 characters and are unique.
- The screenshot TooMany illustrates the result of a “wrong” answer to the previous question where the respondent said there were more 15-year-old people than there were household members. This error is trapped by the program logic in the Qlogic block for QID HH2; for this example, we assume that the Field Interviewer corrected the error when sent back to screen HH2 and moved forward with the interview.
- Q Sequence 10: A start screen should be a Comp type screen so it is not displayed; note that in the Texts table the text for the Start question is the questionnaire version number. The version number is stored there by the designer program.
- Q Sequence 11: QID Fills shows how the GSS set statement is used to create variables used as fills. For example, set {COUNTRYNAME}=“Country” creates a variable that can be used anywhere in the questionnaire and is used as a fill in QID Intro1. See the Texts table for QID Intro1.
- Q Sequence 31: QID Intro1 shows the use of a variable as a fill; the {Agency} variable is filled by its current value as shown in the Intro1 screenshot.
- Q Sequence 41: QID HH1 shows how an if-then syntax is used to manage flow control in a questionnaire. In this screen, if there are no eligible people (i.e., the answer to HH1 is 0), then you would skip to QID NoEligible. HH1 also illustrates the use of a range; in this case a range of 0 to 50 (RangeLo=0 and RangeHi=50). Also, the Special Instructions field for this screen specifies that only integer values will be accepted as answers to HH1. The use of the # sign in the If

statement causes the comparison to be done on a numeric basis versus character if the # sign is not used.

- Q Sequence 51: QID HH2 illustrates the use of three multiple logic statements.
- Q Sequence 111: QID Rloop illustrates the use of the Loop type screen to start a loop over the number of people in the household. The loop should start at 1 (field LoopStart) and go to the values of HH3 (field LoopEnd). For the Loop type screen, the Qnext field tells the GSS what screen starts the loop.
- Q Sequence 131: QID HH4b illustrates the use of numeric comparisons in the Qlogic field. When doing logical comparisons in the GSS, you can do character or numeric comparisons. Use the "#" character before a variable name (e.g., {#HH4b}) to tell the GSS to do numeric compares. If no "#" is used, then the compares are done as character compares (see QID HH4d).
- Q Sequence 201: QID LoopEnd illustrates the use of the LoopEnd type of screen. This screen type demarcates the end of a loop, in this case, the loop over household members. It has two fields that must be filled in: LoopEnd and LoopGoTo. The LoopEnd field is a number or variable that specifies the end limit of the loop. The LoopGoTo is the QID of the screen to go to when the loop counter is less than or equal to this limit (LoopEnd value), and the Qnext field is the QID of the screen to go to when the loop counter is greater than this limit (LoopEnd value).
- Q Sequence 231: QID Select is an example of calling user-supplied subroutines. The Call statements call subroutines that have been added to GSS for GATS. The GatsSelAlgo routine is the sample selection routine and the GATSSaveUserInfo is a special purpose routine that saves intermediate data (e.g., the random number from the sampling routine).
- Q Sequence 261: QID CodeEvents is a Comp type screen that shows an example of how you can auto code events depending on the questionnaire data. Here, if certain data conditions are met, the case is event coded and in addition in other conditions both the HQ case and the IQ case are event coded. The Code_Event and Code_EventIQ are GATS subroutines that write result codes to the relevant GSS data tables.

Exhibit 4-10. GSS Implementation of the GATS Household Questionnaire: Questions Table

Questions													
QSequence	Qid	Qlogic	QType	Special Instructions	QNext	Qanswer	RangeLo	RangeHi	LoopStart	LoopEnd	LoopGoTo	RosterNumber	RosterCol
1	**Start**		**COMP**		**Fills**	**None**	**0**	**0**				**0**	**0**
11	Fills	set {Agency} = "Statistical Agency"; set {CountryName} = "Country"; set {Fill1_0} = "oldest"; set {HH5Flag} = "0"; Set {NoEflag} = "0"; set {CountryExamples_0} = "(FILL APPROPRIATE COUNTRY EXAMPLES: cigarettes, cigars, pipes)";	COMP		Intro	None	0	0				0	0
21	Intro		INFO		Intro1	None	0	0				0	0
31	Intro1	set {HHType} = "{Parm2}"; set {HH5Flag} = "0";	INFO		HH1	None	0	0				0	0
41	HH1	if {#HH1} = 0 then goto NoEligible;	NUM	integer,NODKRE;	HH2	None	0	50				0	0
51	HH2	if {HH2} = "?" or {HH2}="!" then goto HH3; if {#HH2} = 0 then goto NoEligible; If {#HH2} > {#HH1} then goto TooMany; if {HHType} = "Both" then set {HH3} = {HH2}; if {HHType} = "Both" then goto HH4both;	NUM	integer,NODKRE;	HH3	None	0	20				0	0
61	TooMany	backupto HH2;	INFO		toomany	None	0	0				0	0
71	HH3	Set {NoEFlag} = "0"; if {hh3}="?" or {hh3} = "!" then goto NoEligible; if {#HH3} = 0 then goto NoEligible; IF {#HH3} > {#HH2} then goto TooMany2;	NUM	integer,NODKRE;	HH4	None	0	20				0	0
81	TooMany2	backupto HH3;	INFO		TooMany2	None	0	0				0	0
91	HH4		INFO		Rloop	None	0	0				0	0
101	HH4both		INFO		Rloop	None	0	0				0	0
111	Rloop		LOOP		HH4a	None	0	0	1	{HH3_0}		0	0
121	HH4a		TEXT		HH4b	None	0	0				1	0
131	HH4b	if {#HH4b} >= 15 and {#HH4b} <= 17 then goto HH4c;	NUM	integer;NODKRE;	HH4dComp	None	15	110				1	1
141	HH4c		LIST		HH4cYear	MONTHS	0	0				1	2

(continued)

Exhibit 4-10. GSS Implementation of the GATS Household Questionnaire: Questions Table (Page 2 of 4)

Questions													
QSequence	Qid	Qlogic	QType	Special Instructions	QNext	Qanswer	RangeLo	RangeHi	Loop Start	Loop End	Loop GoTo	Roster Number	Roster Col
151	HH4cYear	if {HH4c} = "77" or {HH4c} = "99" or {HH4cYear} = "?" or {HH4cYear} = "!" or {HH4cYear} = "7777" or {HH4cYear} = "9999" then goto HH4d; call ValidateBDay; if {#calcyears} < 15 then goto ValidateAge;	NUM	Integer,NODKRE, RngInclude=7777; 9999;	HH4dComp	None	1900	2000				1	3
161	ValidateAge	backupto HH4c;	INFO		ValidateAge	None	0	0				0	0
165	HH4dComp	set {MFFill_0} = "[RECORD GENDER (FOR VERIFICATION IF NECESSARY)]"; if {HHType_0} = "Both" then set {MFFill_0} = "Is this person male or female?";	Comp		HH4d	None	0	0				0	0
171	HH4d	IF {HHType_0} = "Both" then goto HH4e; if {HH4d} ="1" and {HHType_0} = "Female" then goto WrongGender; if {HH4d} = "2" and {HHType_0} = "Male" then goto WrongGender;	LIST		HH4e	MALEFEM	0	0				1	4
181	WrongGender	backupto HH4d;	INFO		WrongGender	None	0	0				0	0
191	HH4e	set {Fill1_0}="next oldest"	LIST		LoopEnd	YESNO?!	0	0				1	5
201	LoopEnd		LEND		EditRosterIntro	None	0	0	1	{HH3_0}	HH4a	0	0
211	EditRosterIntro		INFO		SELECT	YESNO	0	0				0	0
231	Select	set {HH5Flag} = "0"; call GatsSelAlgo; call GATSSaveUserInfo	COMP		HH5	None	0	0				0	0
241	HH5	set {HH5Flag}="1";	INFO	NoBackup;	CodeEvents	None	0	0				0	0
251	NoEligible	Set {NoEFlag}="1";	INFO		CodeEvents	None	0	0				0	0
261	CodeEvents	if {HH5Flag} = "1" then set {EventCode} = "200"; if {NoEflag} = "1" then set {EventCode} = "201"; if {HH5Flag} = "1" then set {EventComment} = "Screener Complete"; if {NoEflag} = "1" then set {EventComment} = "Screener Complete No Eligibles"; call Code_Event; if {NoEflag} = "1" then set {EventCode} = "401"; if {NoEflag} = "1" then set {EventComment} = "HH No Eligibles"; if {NoEflag} = "1" then call Code_EventIQ;	Comp		END	None	0	0				0	0
271	END		END		Null	None	0	0				0	0

Exhibit 4-10. GSS Implementation of the GATS Household Questionnaire: Questions Table (Page 3 of 4)

Texts				
Pkey	**Qid**	**Language**	**QText**	**Qwav**
255	Select	English		
256	END	English	DO YOU REALLY WANT TO END THE INTERVIEW? IF SO, TAP Exit TO END, OTHERWISE TAP Back TO GO BACK.	
349	WrongGender	English	[YOU SHOULD BE ROSTERING ONLY {HHType_0}s; PLEASE DOUBLE CHECK THIS DATA.]	
350	HH4e	English	Does this person currently smoke tobacco, including {CountryExamples_0}?	
351	LoopEnd	English		
376	EditRosterIntro	English	[IF YOU NEED TO REVIEW THE ROSTER, SELECT “ROSTER” FROM THE TOOLS MENU. TAP THE BACK BUTTON IF YOU NEED TO MAKE CHANGES. TAP THE NEXT BUTTON TO SELECT THE RESPONDENT.]	
380	NoEligible	English	[THERE ARE NO ELIGIBLE HOUSEHOLD MEMBERS. THANK THE RESPONDENT FOR HIS/HER TIME. THIS WILL BE RECORDED IN THE RECORD OF CALLS AS A CODE 201.]	
386	CodeEvents	English		
387	HH5	English	[NAME OF THE SELECTED ELIGIBLE PERSON IS: {SelName} ASK IF THE SELECTED RESPONDENT IS AVAILABLE AND IF SO, PROCEED TO THE *INDIVIDUAL QUESTIONNAIRE*. IF THE SELECTED RESPONDENT IS NOT AVAILABLE, MAKE AN APPOINTMENT AND RECORD IT AS A COMMENT ON RECORD OF CALLS.]	
399	TooMany2	English	[YOU CAN'T/SHOULDN'T HAVE MORE {HHtype}s 15 THAN THERE ARE TOTAL HH MEMBERS >= 15. PLEASE DOUBLE CHECK THE ANSWERS SO FAR.]	
400	HH4	English	I now would like to collect information about the {HH3} {HHType}s that live in this household who are 15 years of age or older. Let’s start listing the {HHType}s from oldest to youngest.	
401	HH4both	English	I now would like to collect information about only these persons that live in this household who are 15 years of age or older. Let’s start listing them from oldest to youngest.	
402	Rloop	English		
412	HH4dComp	English		
416	ValidateAge	English	[AGE CALCULATED FROM BIRTH MONTH AND BIRTH YEAR IS LESS THAN 15. PLEASE DOUBLE CHECK THESE ANSWERS.]	

(continued)

Exhibit 4-10. GSS Implementation of the GATS Household Questionnaire: Questions Table (Page 4 of 4)

Texts				
Pkey	**Qid**	**Language**	**QText**	**Qwav**
417	HH4a	English	What is the {Fill1_0} person's first name?	
418	HH4b	English	What is this person's age? [IF RESPONDENT DOESN'T KNOW, PROBE FOR AN ESTIMATE]	
419	HH4c	English	What is the month of this person's date of birth?	
420	HH4cYear	English	What is the year of this person's date of birth? [IF DON'T KNOW, ENTER 7777 IF REFUSED, ENTER 9999]	
421	HH4d	English	{mffILL_0}	
422	Start	English	Ver.: 2010 07 19 035	
423	Fills	English		
424	Intro	English	[THE HOUSEHOLD SCREENING RESPONDENT MUST BE 18 YEARS OF AGE OR OLDER AND YOU MUST BE CONFIDENT THAT THIS PERSON CAN PROVIDE ACCURATE INFORMATION ABOUT ALL MEMBERS OF THE HOUSEHOLD. IF NEEDED, VERIFY THE AGE OF THE HOUSEHOLD SCREENING RESPONDENT TO MAKE SURE HE/SHE IS 18 YEARS OF AGE OR OLDER.]	
425	Intro1	English	An important survey of adult tobacco use behavior is being conducted by the {Agency} throughout {CountryName} and your household has been selected to participate. All houses selected were chosen from a scientific sample and it is very important to the success of this project that each participates in the survey. All information gathered will be kept strictly confidential. I have a few questions to find out who in your household is eligible to participate.	
426	HH1	English	First, I'd like to ask you a few questions about your household. In total, how many persons live in this household? [INCLUDE ANYONE WHO CONSIDERS THIS HOUSEHOLD THEIR USUAL PLACE OF RESIDENCE]	
427	HH2	English	How many of these household members are 15 years of age or older?	
428	TooMany	English	[YOU CAN'T/SHOULDN'T HAVE MORE PEOPLE >= 15 YEARS OLD THAN THERE ARE TOTAL HH MEMBERS; PLEASE DOUBLE CHECK THE ANSWERS SO FAR.]	
429	HH3	English	How many {HHType} household members are 15 years of age or older?	

Exhibit 4-11 lists the answer sets that are specified in the Answers table. These are the list options that are used in List or ATA (All That Apply) type screens. The UID column is an auto number field that serves as the primary key (an unique numeric ID for each row). It is automatically generated. The AID column is the ID of the answer set or its name; the Asequence column tells GSS what order to display the possible answer choices; the Acode column is the code that is stored in the database (in the responses table) when an answer choice is selected; the Atext column is the text displayed; and the Language column identifies the language of this answer set. Here, four answer sets are defined with AIDs: MaleFem, Months, YesNo, and YesNO?!. These are arbitrary names for the answer sets and they must be unique within a questionnaire specification. Answer sets can be used on more than one question or none at all. (If you use sets from prior questionnaire building, you can carry them over even if you do not use them.) The answer sets are used in the following screens:

- HH4d uses the MaleFem answer set.
- HH4e uses the YesNo?! answer set.
- HH4c uses the Months answer set.

Exhibit 4-11. GSS Implementation of the GATS Household Questionnaire: Answers Table

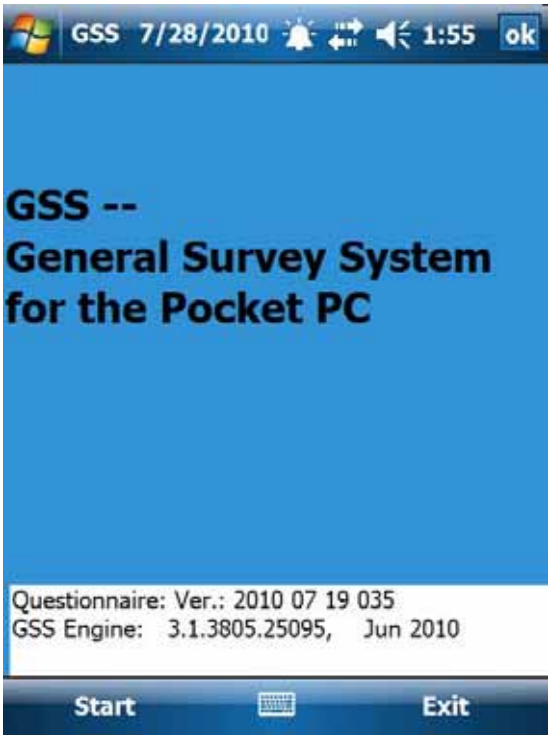
Answers						
UID	**AID**	**Language**	**ASequence**	**ACode**	**AText**	**AWav**
338	YESNO	English	1	1	YES	
339	YESNO	English	2	2	NO	
412	YESNO?!	English	1	1	YES	
413	YESNO?!	English	2	2	NO	
414	YESNO?!	English	3	?	DON'T KNOW	
415	YESNO?!	English	4	!	REFUSED	
464	MONTHS	English	1	1	01	
465	MONTHS	English	2	2	02	
466	MONTHS	English	3	3	03	
467	MONTHS	English	4	4	04	
468	MONTHS	English	5	5	05	
469	MONTHS	English	6	6	06	
470	MONTHS	English	7	7	07	
471	MONTHS	English	8	8	08	
472	MONTHS	English	9	9	09	
473	MONTHS	English	10	10	10	
474	MONTHS	English	11	11	11	
475	MONTHS	English	12	12	12	
476	MONTHS	English	13	77	DON'T KNOW	
477	MONTHS	English	14	99	REFUSED	
478	MALEFEM	English	1	1	MALE	
479	MALEFEM	English	2	2	FEMALE	

The other screens do not use answer sets, because they are not List or ATA type screens. The Awav column is not used in GATS. Since this example is using only English, the Answers table shows only one language. If multiple languages were used, then each language would have a set of rows with the same data items but the text items in the desired language. For example if the MaleFem answer set was to be in French, we would also see the following rows in the table:

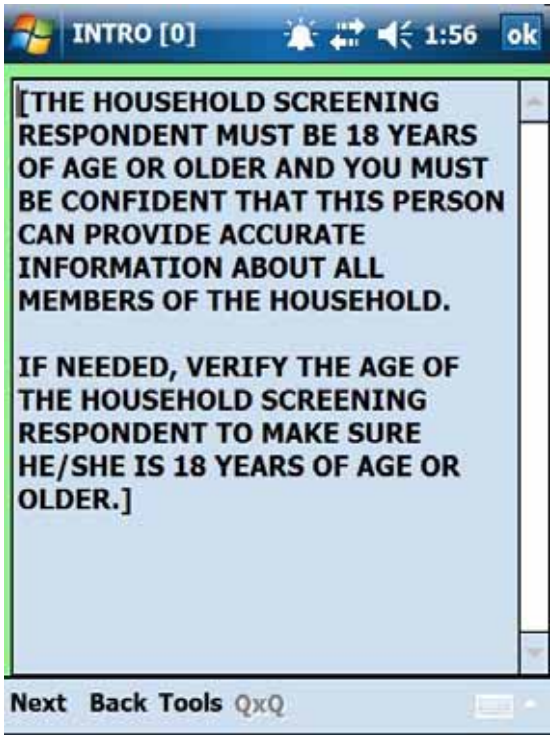
900	MALEFEM	French	1	1	Homme
901	MALEFEM	French	2	2	Femme

The specifications provided above lead to the following set of screens for the GATS *Household Questionnaire*, given the specific answers shown on the screens. This example assumes the “Type” variable that specifies the household sampling level is “Both.” Other screens are skipped due to the routing path dictated by the answers.

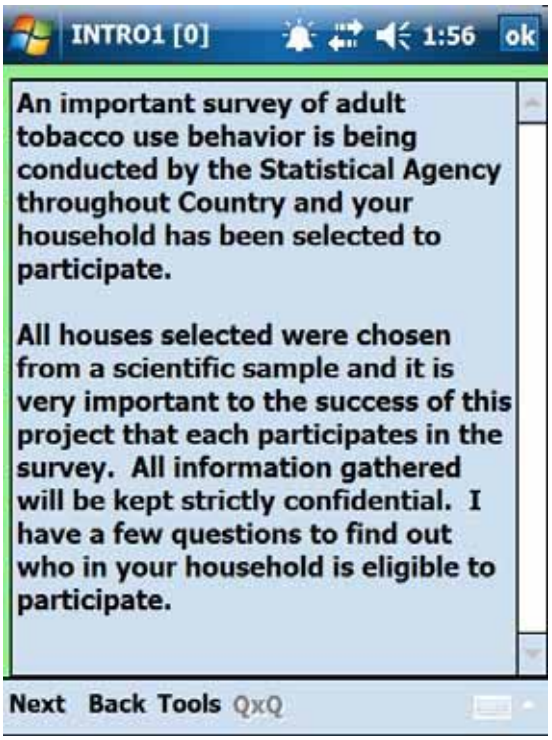
GSS Start Screen



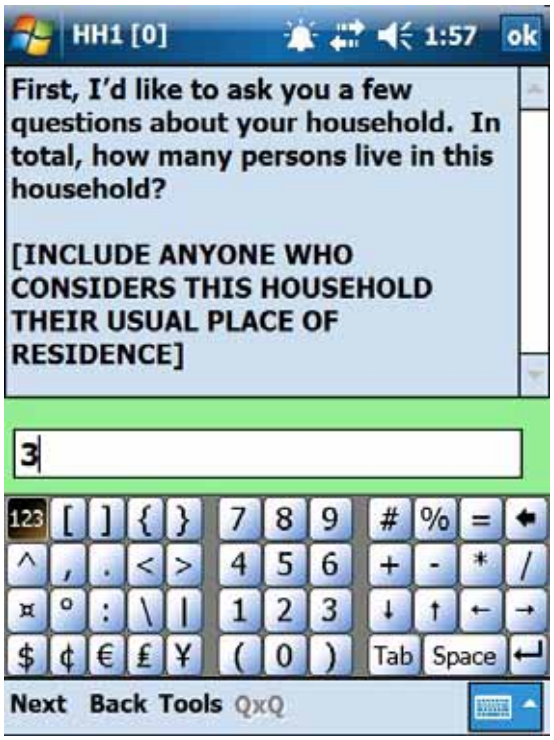
Intro Screen



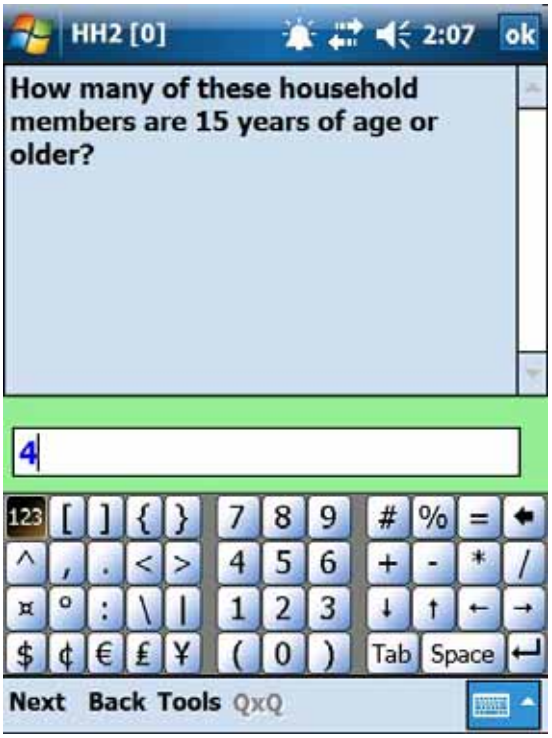
Intro1 Screen



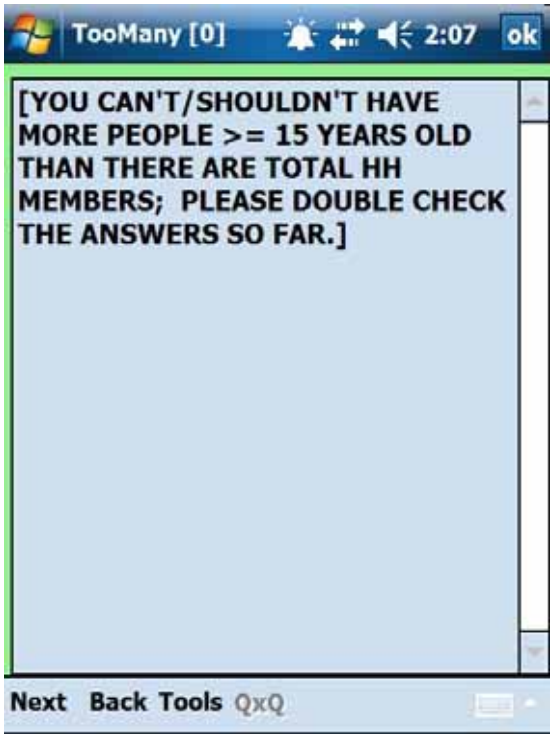
HH1 Screen



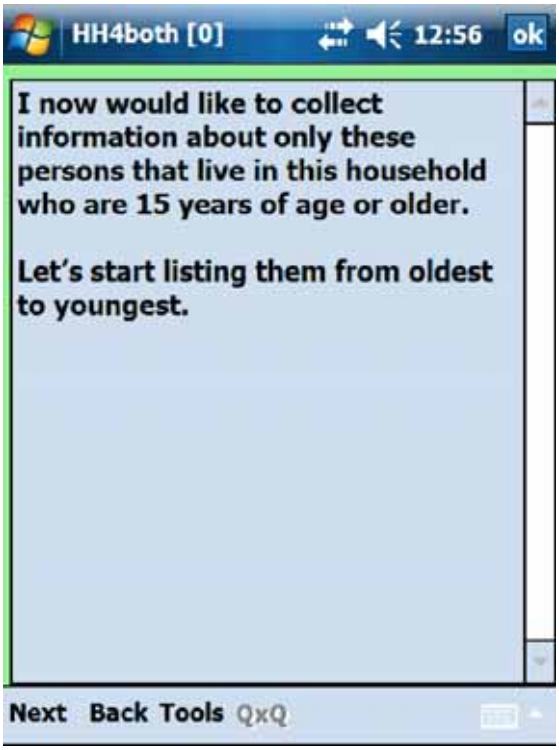
HH2 Screen



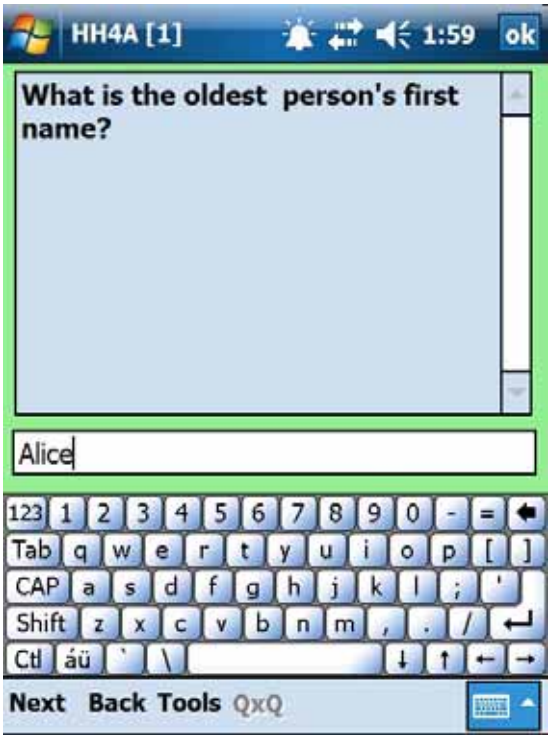
TooMany Screen



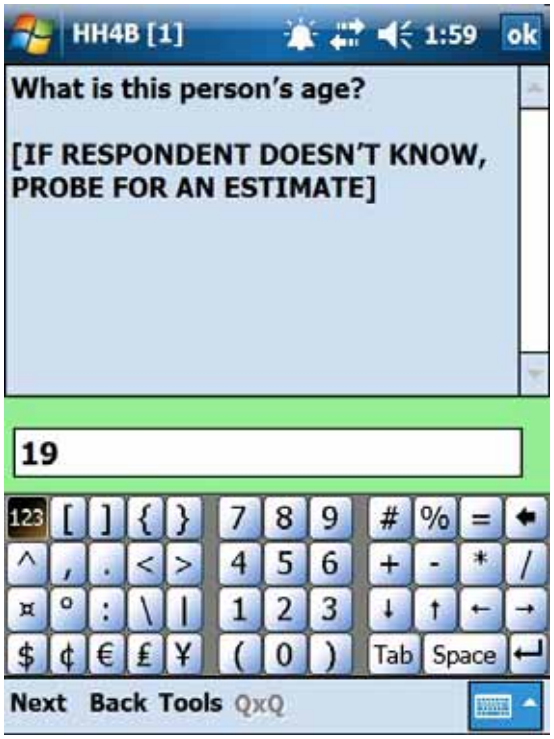
HH4both Screen



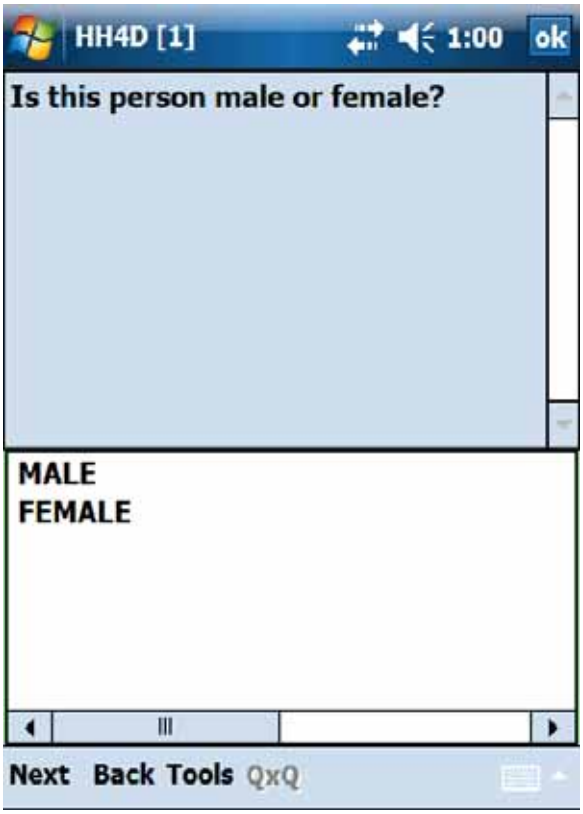
HH4a Screen



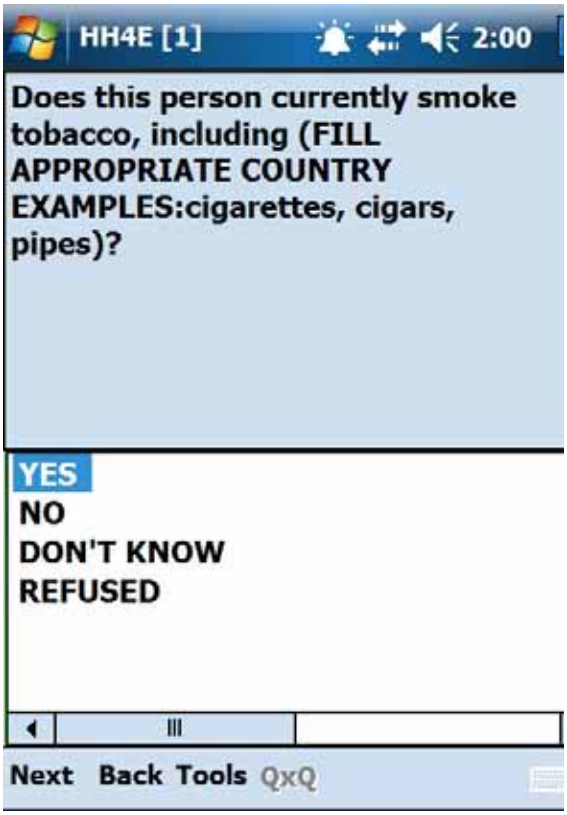
HH4b Screen



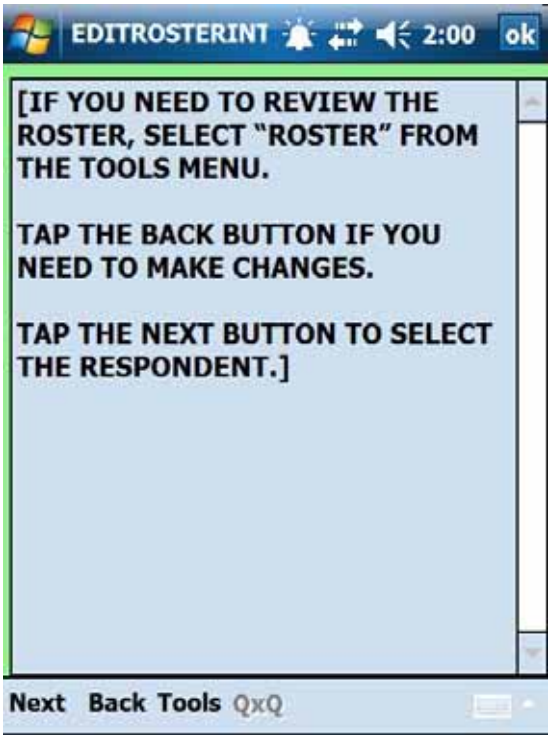
HH4D Screen



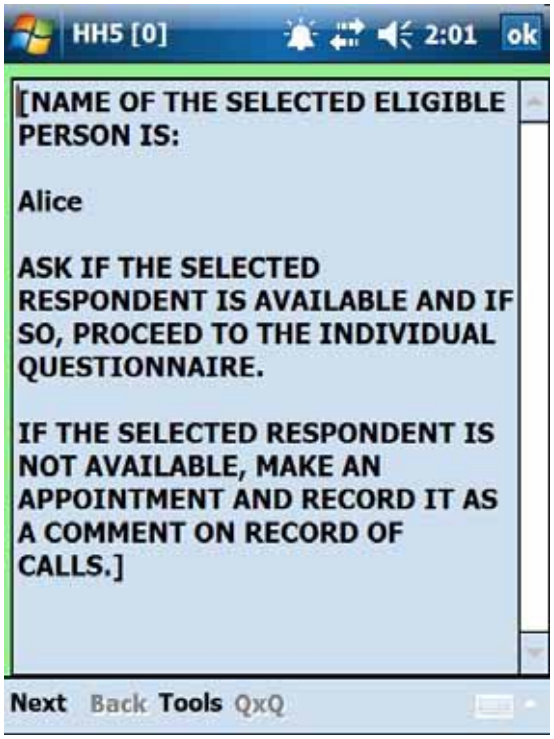
HH4E Screen



EditRosterIntro



HH5 Screen



End Screen

5. GSS Developer's Tool Set

This chapter explains the functionality of a set of software products that support the development of forms in GSS. The GSS Developer's Tool Set is a control program that serves as the main menu or pathway to the suite of GSS tools.

5.1 GSS Developer's Tool Set

The GSS Developer's Tool Set organizes PC-based GSS modules that support questionnaire development and data management into a single user interface to make access and use of these programs easier and simpler, **Exhibit 5-1** shows the initial or startup view.

The modules include:

- the Questionnaire Designer (Quex Designer), the GSS Survey Designer, assists the user in preparing survey instruments. Designer allows for design of surveys using multiple languages;
- a tool to review and edit GATS MDB tables (Edit DB Tables);
- tools to allow users to create and edit GATS "case" files (Case File Tools);
- data aggregation tools that allow GATS users to manage, aggregate, and report on handheld data files (Data Aggregation);
- a utility to convert MS Access MDB files to Pocket PC compatible format SDF files (Make SDF files); and
- a Help module that contains help files and the major GATS GSS documents.

The general design strategy for this interface to the GSS suite of tools is to make it quite GATS specific and oriented toward GATS users.

5.1.1 Platform and Development Language

The systems were developed on standard PC platforms running Windows XP Professional Version 2003 Service Pack 3. The Tool Set is designed for standard size PC laptops with screen resolution at or below 1024 x 768. The programs were developed using Microsoft Visual Studio 2005 in Visual Basic. The .Net framework modules 2.0 or later are used and need to be on any target platform.

5.1.2 Main Menu Module

A sample of the start screen for the Main Menu is shown in **Exhibit 5-1**.

Exhibit 5-1. Start Screen of Main Menu for GSS Developer Tools

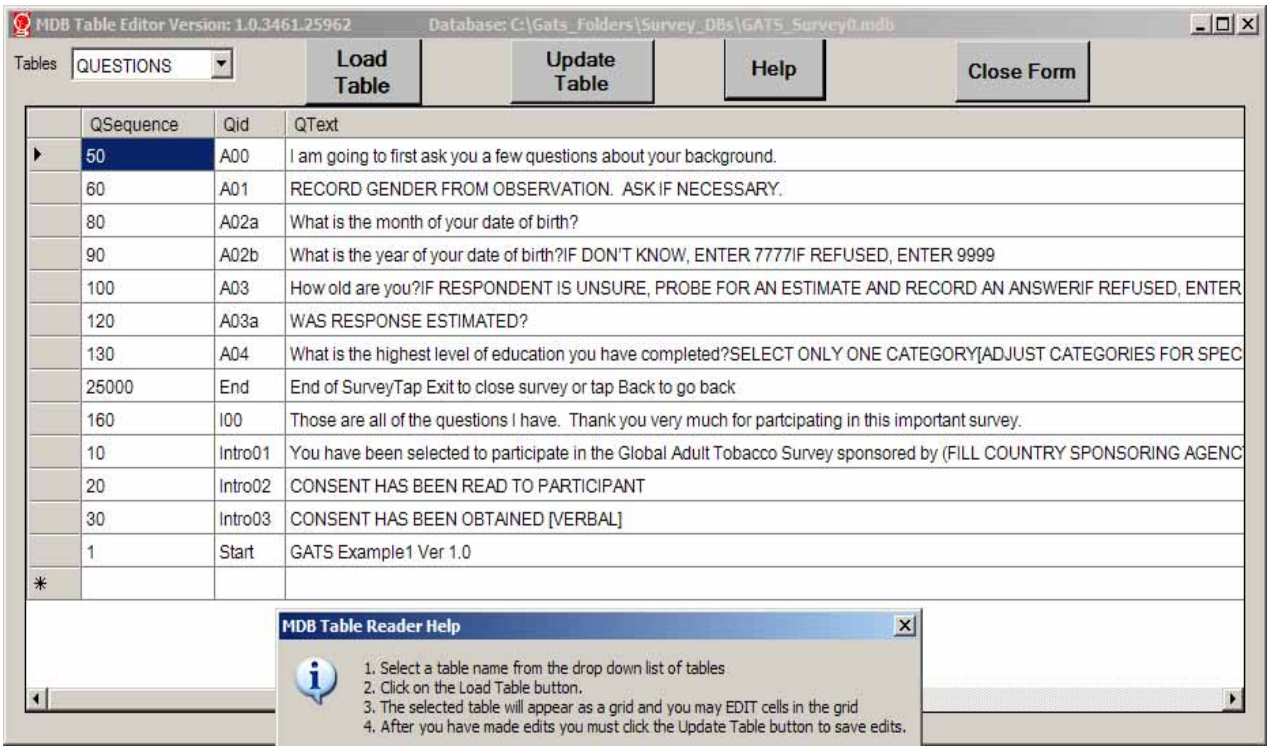
Eight main menu choices invoke the GSS support programs. The main menu items and their submenus are as follows:

- **Quex Designer**—invokes the Questionnaire Designer to work on either Gats_Survey0 or Gats_Survey1 or a new survey (see **Section 5.2** for a detailed description of this tool)
  - HQ (Survey0)
  - IQ (Survey1)
  - New Survey Questionnaire
- **Edit DB Tables**—allows the user to edit database tables associated with Survey 0, Survey 1, or the CMSDB databases
  - HQ (Survey0)
  - IQ (Survey1)
  - CMS DB Tables
- **Case File Tools**—allows users to edit and update GATS case files and design the layout of the CMS case grid
  - Edit Case File
  - Import Items from CSV
  - CMS Grid Designer
- **Data Aggregation**—provides tools to manage, view, aggregate, and report on handheld data (see **Section 5.3** for a detailed description of these tools)

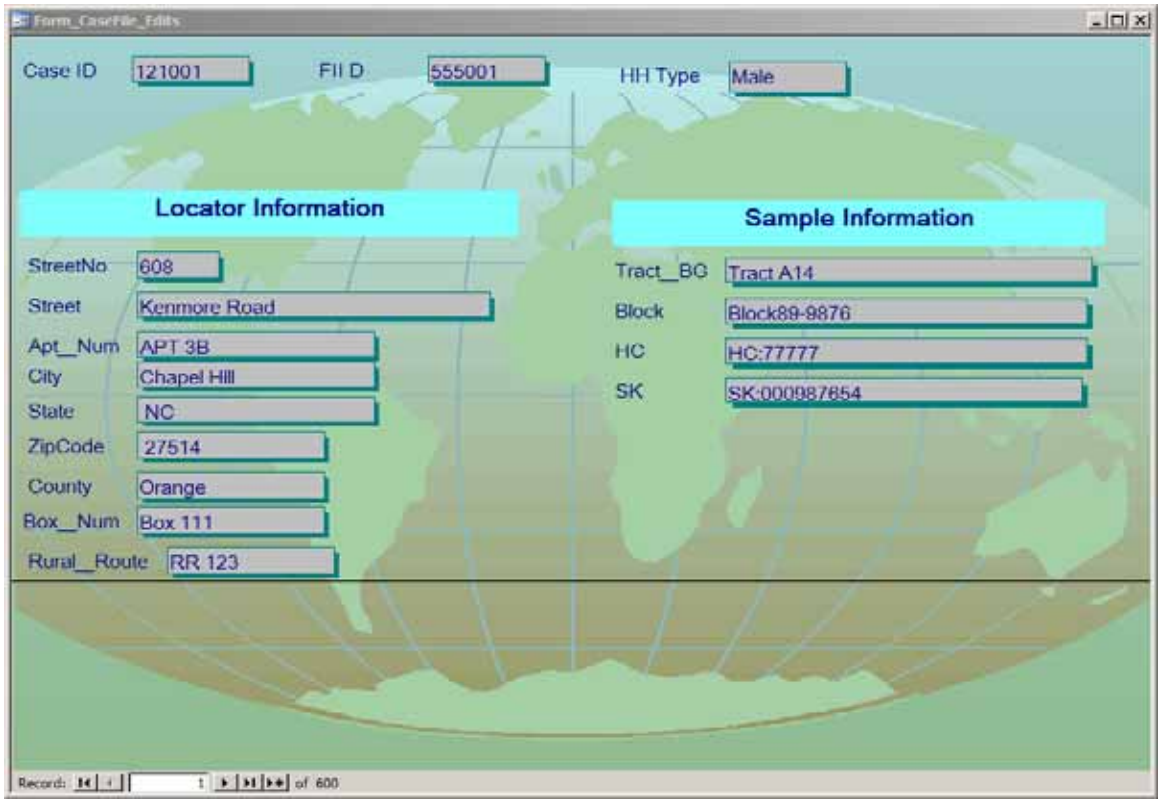
  - View Data
  - Aggregate Data
  - Export to CSV
  - Transpose Data
  - Reporting
  - Master File Merge
- **Make SDF Files**—converts one of the survey MDBs or all three to SDF format
  - HQ (Survey0)
  - IQ (Survey1)
  - CMSDB
  - All three databases
- **Help**—provides access to GSS documents (GSS Guide, GATS Field Interviewer and Field Supervisor Manuals, Data Management Implementation Manual, List of Day Codes)
- **Program Options**—allows users to set up paths, passwords, and other required parameters for programs in one location and save them to be passed in to programs as needed
  - Setup parameters
- **Exit**—exits the program

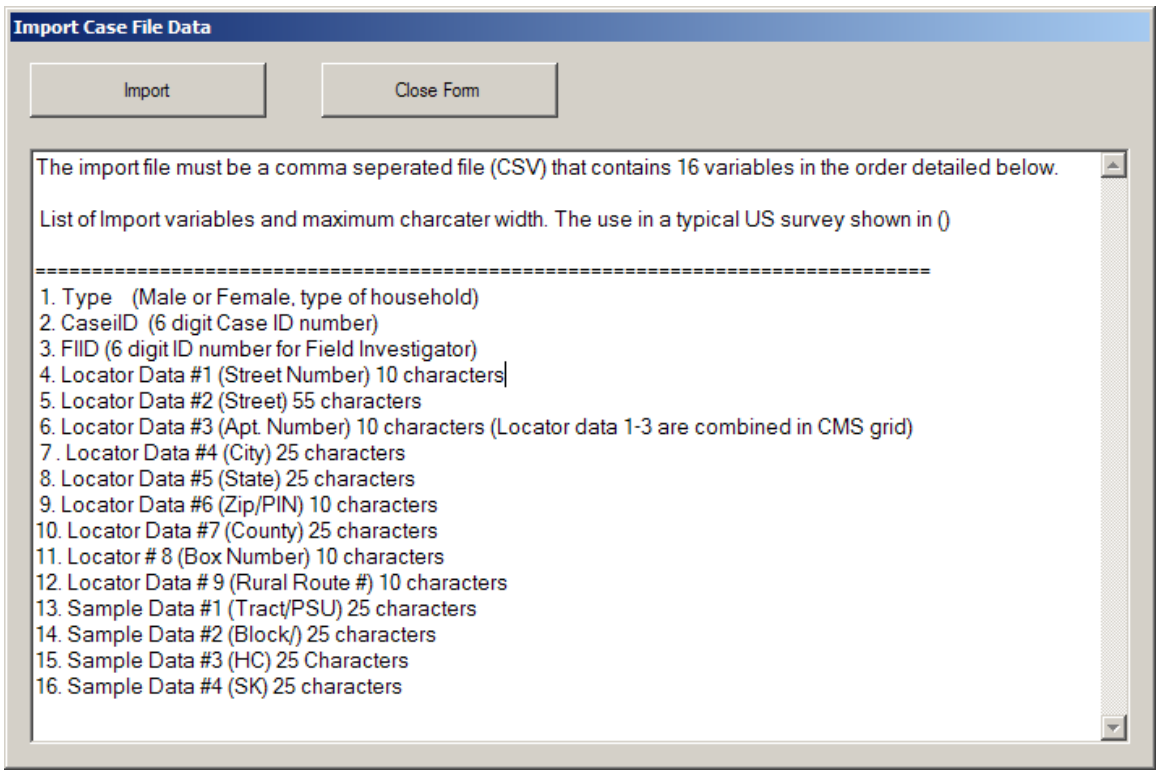
The menu system allows the GATS developer to access the GSS suite of tools from one location. The developer can specify the locations of key folders one time and save that definition for the next use of the tool. Each menu and its function are described in the following list.

- **Quex Designer**—initiates the Designer program with a specific survey form (HQ, IQ, or New Survey Questionnaire) with the file path as specified by the setup parameters of the main menu. If you choose new survey questionnaire, you will be asked to browse and locate the MDB file you wish to employ. See **Section 5.2** for a detailed introduction to the Designer tool.
- **Edit DB Tables**—allows developers to directly review and edit internal tables in the Survey 0 and 1 (Tables: Answers, INI, Questions, Texts, QxQ) databases and the CMSDB (Tables: EvtDef, INI, Messages, TrainingCases). The following screenshot shows the Questions table open for Survey 0 and the help file open as well. The user can edit the grids for all tables and add new rows for all tables except the Answers and TrainingCases tables. The Help screen in the following screenshot explains how to operate the edit options.



- **Case File Tools**—allows GATS developers to work with the GATS case file. You can edit a case file, import items into a case file from a CSV file, export items from the case file to a CSV file or use the CMS Grid designer to control the order width and columns that display in the CMS case grid.
- **Edit Case File**—the Edit Case File option brings up an access form (for this option to work, users must have MS Access 2003 or later installed on their PC) that allows the user to edit a template of 600 cases that could form the base for a case file. This is an easy way to edit and manage a small case file. If you need a bigger file, you can use the Import Items from CSV option to bring in a file of updates or new records to add to the case file. The format and requirements for the import file are listed on the Import Case File Data screen on the next page.


Form_CaseFile_Edits
Case ID
121001
FI I D
555001
HH Type
Male
Locator Information
Sample Information
StreetNo
608
Street
Kenmore Road
Apt_Num
APT 3B
City
Chapel Hill
State
NC
ZipCode
27514
County
Orange
Box_Num
Box 111
Rural_Route
RR 123
Tract_BG
Tract A14
Block
Block89-9876
HC
HC:77777
SK
SK:000987654
Record:
of 600


Import Case File Data
Import
Close Form
The import file must be a comma seperated file (CSV) that contains 16 variables in the order detailed below.
List of Import variables and maximum charcater width. The use in a typical US survey shown in ()
==========
1. Type (Male or Female, type of household)
2. CaseiID (6 digit Case ID number)
3. FIID (6 digit ID number for Field Investigator)
4. Locator Data #1 (Street Number) 10 characters
5. Locator Data #2 (Street) 55 characters
6. Locator Data #3 (Apt. Number) 10 characters (Locator data 1-3 are combined in CMS grid)
7 . Locator Data #4 (City) 25 characters
8. Locator Data #5 (State) 25 characters
9. Locator Data #6 (Zip/PIN) 10 characters
10. Locator Data #7 (County) 25 characters
11. Locator # 8 (Box Number) 10 characters
12. Locator Data # 9 (Rural Route #) 10 characters
13. Sample Data #1 (Tract/PSU) 25 characters
14. Sample Data #2 (Block/) 25 characters
15. Sample Data #3 (HC) 25 Characters
16. Sample Data #4 (SK) 25 characters

Exporting a Case File

- Once a case file exists in Access, you can use the Access export options to create a tab delimited txt file that GSS can use to input cases for iPAQs. The steps are:

Step 1: Navigate to the **tables menu** of Access.

Step 2: Highlight the TmpCasefile table or the table you wish to export.

Step 3: Choose the **File** menu.

Step 4: Choose the **Export** submenu of the **File** menu.

Step 5: Select a location to store output file and chose the file type as text files (*.txt; *.csv; *.tab; *.asc).

Step 6: Click **Export**.

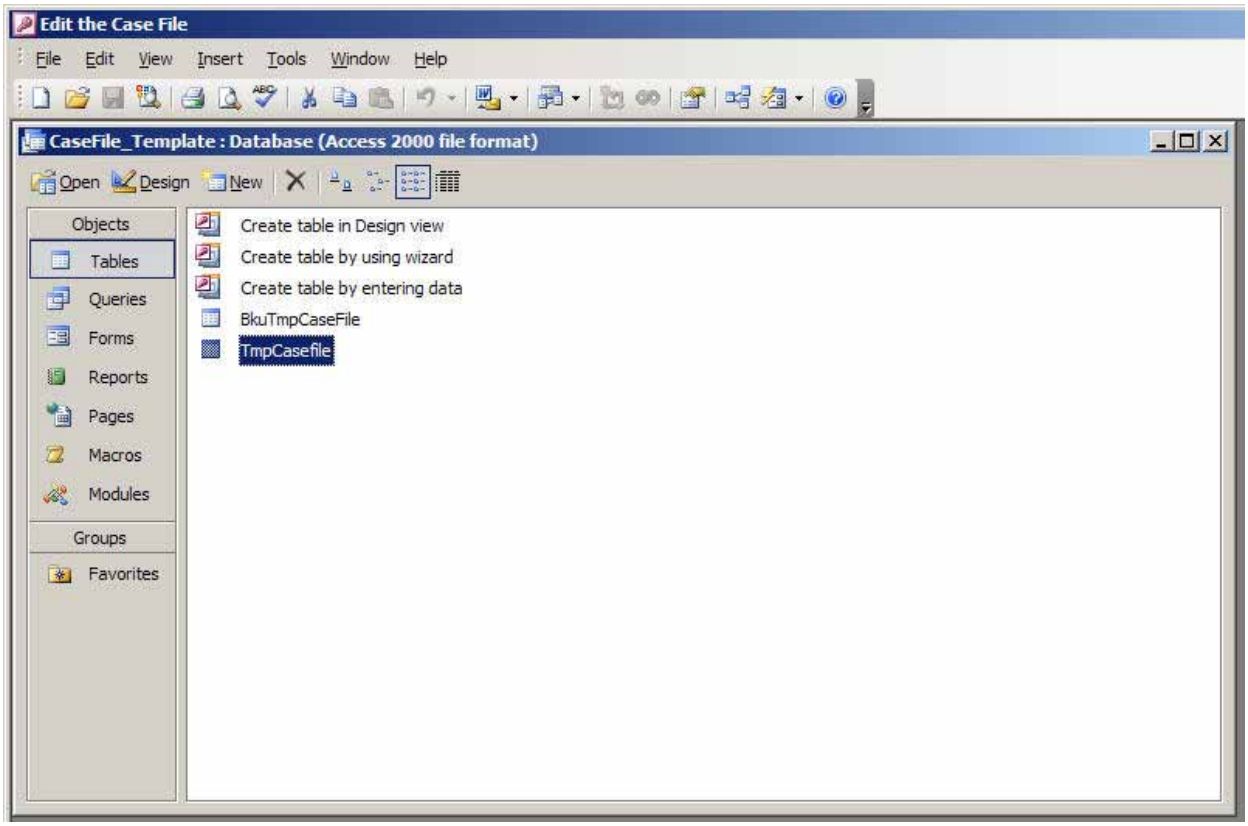
Step 7: Click **Advanced** … and chose the following options:

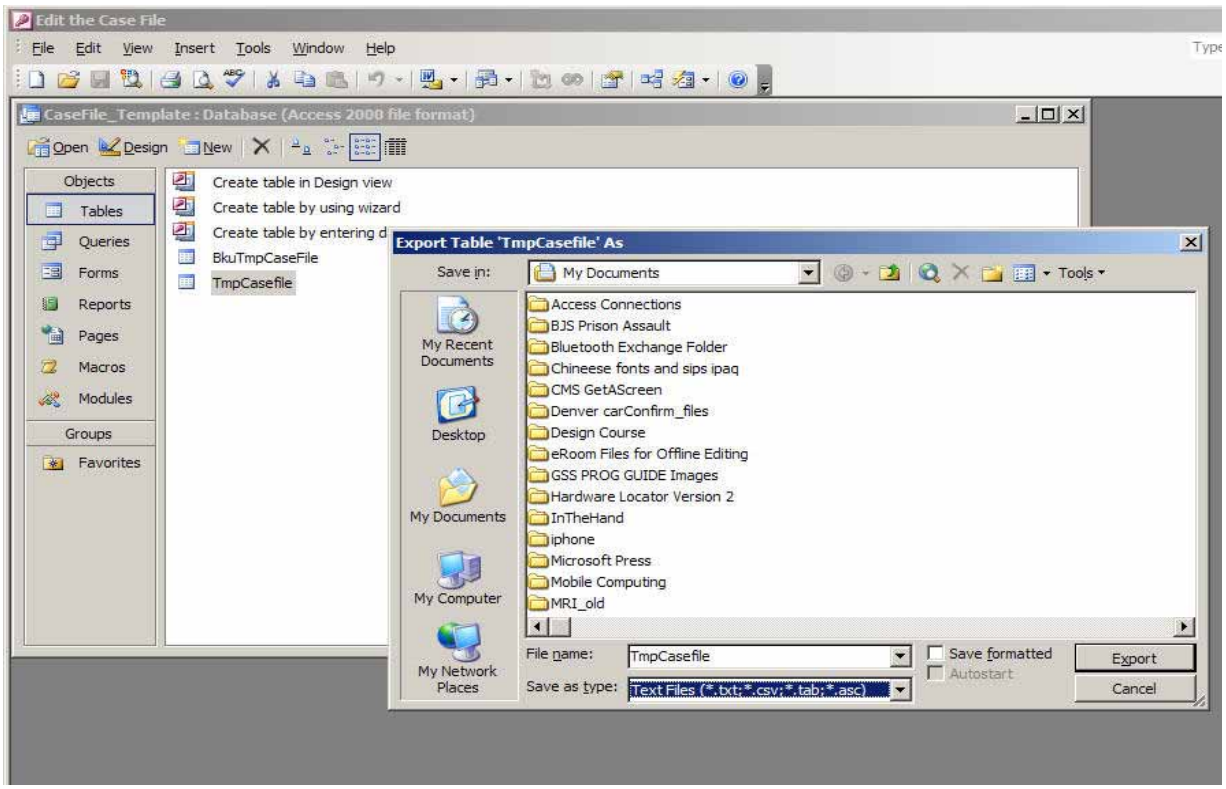
- code page: UNICODE
- field delimiter: {tab}
- text qualifier: {none}

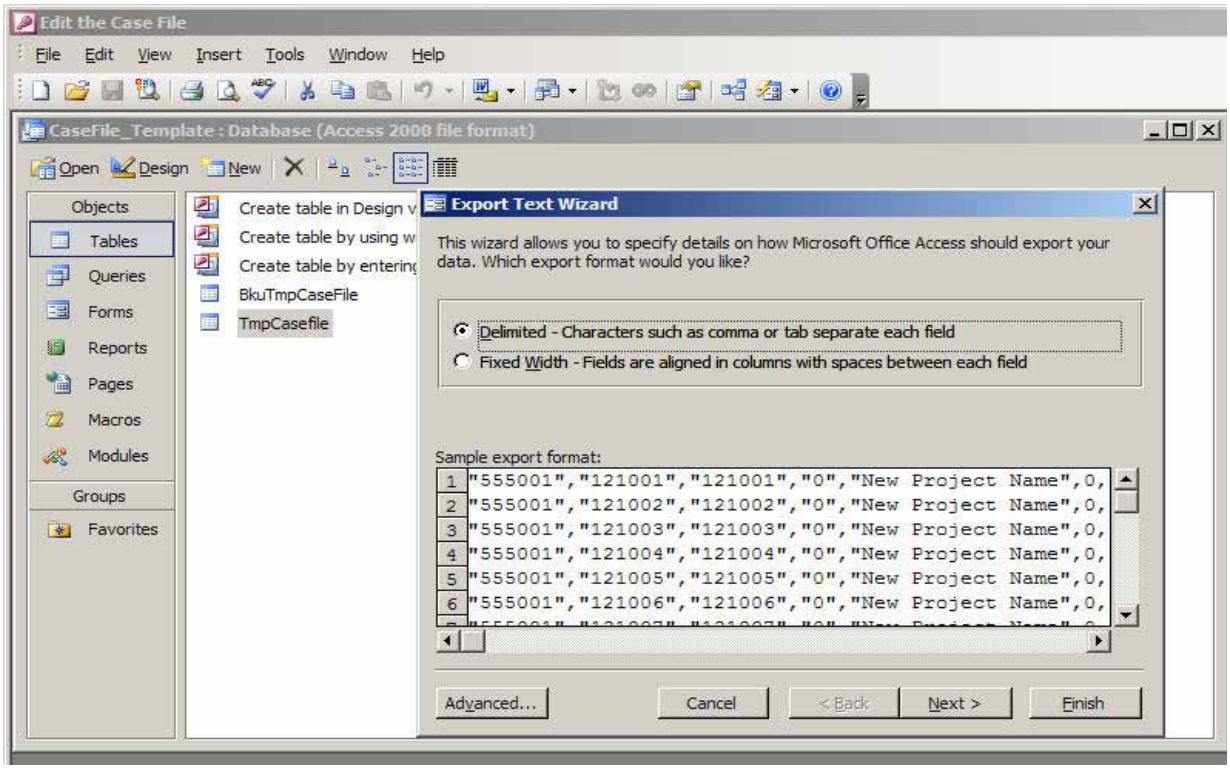
Step 8: Click OK to close the advanced screen.

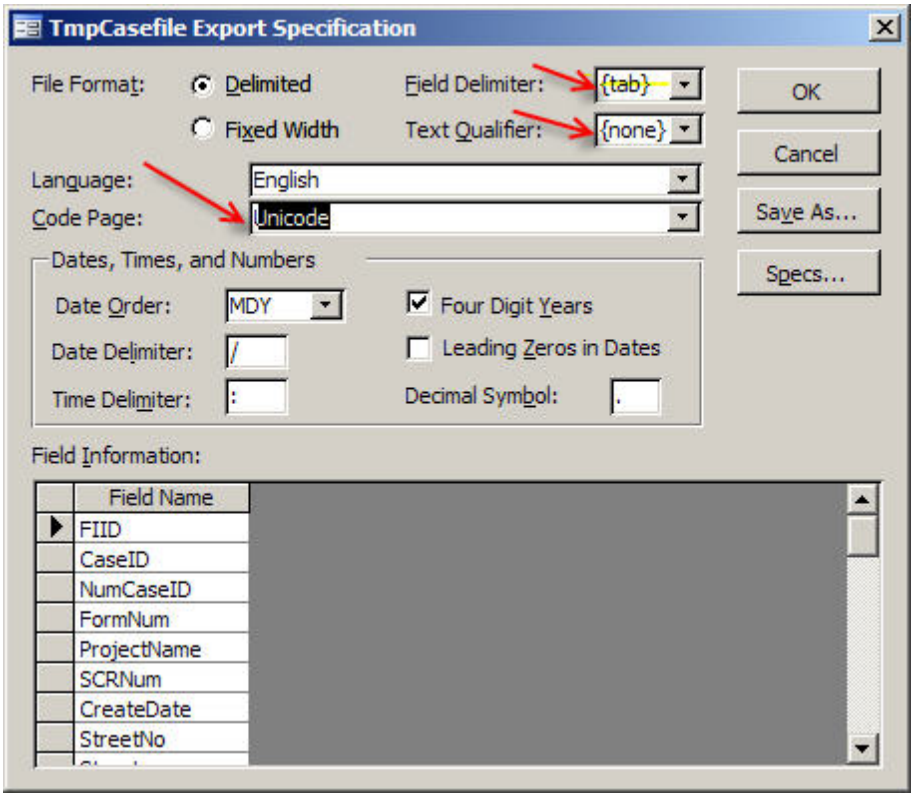
Step 9: Click **Finish** to complete the export.

The sequence of screenshots below shows the steps and option choices.




Edit the Case File
File Edit View Insert Tools Window Help
CaseFile_Template : Database (Access 2000 file format)
Open Design New
Objects
Tables
Queries
Forms
Reports
Pages
Macros
Modules
Groups
Favorites
Create table in Design view
Create table by using wizard
Create table by entering d
BkuTmpCaseFile
TmpCasefile
Export Table 'TmpCasefile' As
Save in:
My Documents
Tools
My Recent Documents
Desktop
My Documents
My Computer
My Network Places
Access Connections
BJS Prison Assault
Bluetooth Exchange Folder
Chineese fonts and sips ipaq
CMS GetAScreen
Denver carConfirm_files
Design Course
eRoom Files for Offline Editing
GSS PROG GUIDE Images
Hardware Locator Version 2
InTheHand
iphone
Microsoft Press
Mobile Computing
MRI_old
File name:
TmpCasefile
Save as type:
Text Files (*.txt;*.csv;*.tab;*.asc)
Save formatted
Autostart
Export
Cancel


Edit the Case File
File Edit View Insert Tools Window Help
CaseFile_Template : Database (Access 2000 file format)
Open Design New
Objects
Tables
Queries
Forms
Reports
Pages
Macros
Modules
Groups
Favorites
BkuTmpCaseFile
TmpCasefile
Export Text Wizard
This wizard allows you to specify details on how Microsoft Office Access should export your data. Which export format would you like?
Delimited - Characters such as comma or tab separate each field
Fixed Width - Fields are aligned in columns with spaces between each field
Sample export format:
1 "555001","121001","121001","0","New Project Name",0,
2 "555001","121002","121002","0","New Project Name",0,
3 "555001","121003","121003","0","New Project Name",0,
4 "555001","121004","121004","0","New Project Name",0,
5 "555001","121005","121005","0","New Project Name",0,
6 "555001","121006","121006","0","New Project Name",0,
Advanced...
Cancel
< Back
Next >
Finish

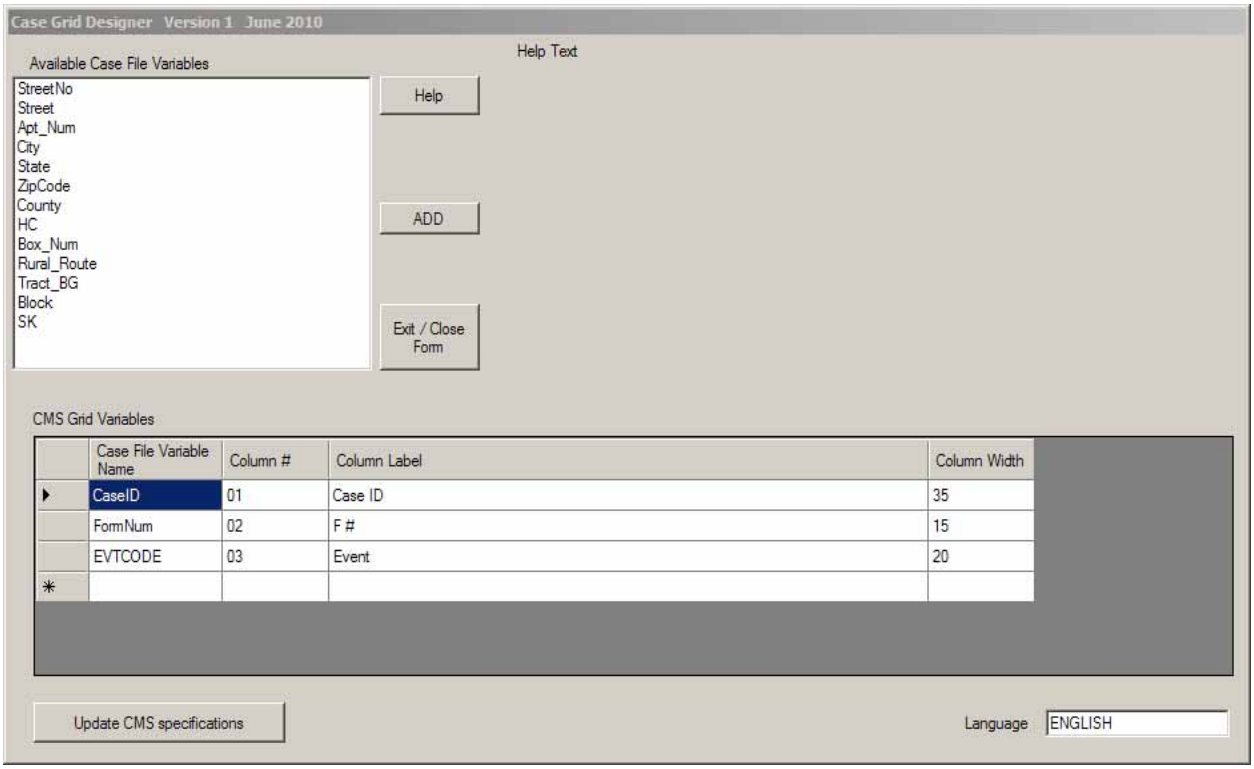


CMS Grid Designer

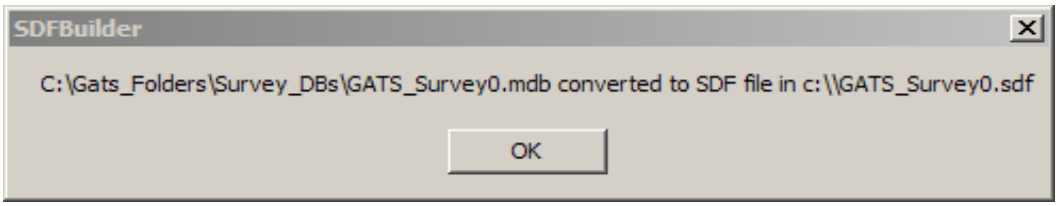
This utility allows users to control the way in which the data in the case file are displayed on the CMS case grid. The user can select which columns from the case file are to be displayed, their width, the order of the columns, and the header text. **Exhibit 5-2** shows the first screen for the grid designer. It shows three case grid variables already selected and in the grid (CaseID, FormNum, and EvtCode). These three variables should be kept but the user can change the width, the order, and the column text of the Formnum and Evtcode variables. If you wish to add additional columns to the case grid display, highlight the desired variable in the Available Case File Variable list box and then tap the **ADD** button. To change the order of the columns, change the column number entries and sort the list by tapping the **Column #** header on the CMS Grid Variables grid. If you want to change the column labels or widths, edit the grid cells directly. If you wish to remove a variable, highlight the row in the CMS Grid Variables grid and use the keyboard delete key to delete the row. The Language text box at the bottom of the screen is for the Language name of the text you will enter as column headings. If you are entering English labels, do not change the Language text box default of ENGLISH. If you are entering text in another language, enter it in all capital letters, for example, SPANISH or MANDARIN. This utility will update the messages table for the specified language to update the labels you have changed and update the INI table to show the selections you have made. A help button for this utility provides instructions.

When you have made your selections, tap the **Update CMS specifications** button and the CMSDB.mdb tables will be updated to implement your choices.

Exhibit 5-2. CMS Grid Designer: First Screen



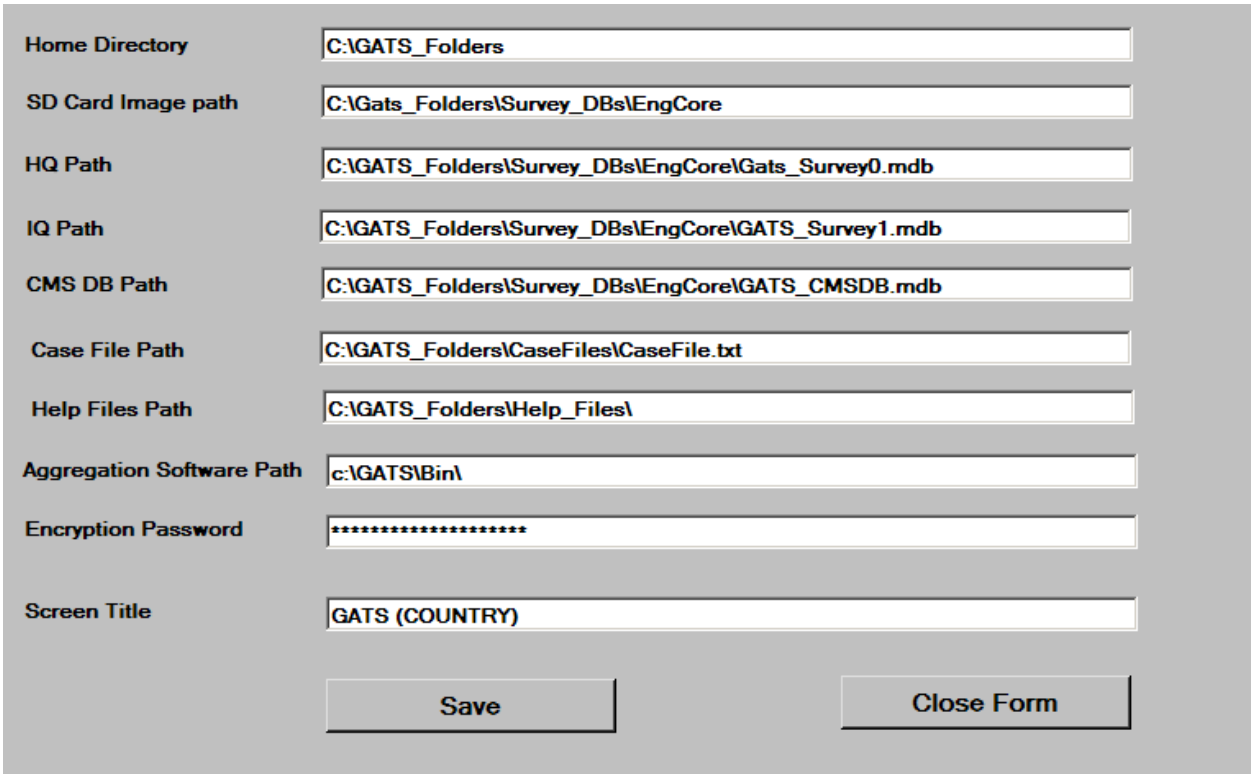
- **Data Aggregation**—brings up the data aggregation program and/or one of its suboptions. In addition it brings up data management and reporting tool options. See **Section 5.3** of this chapter for a detailed discussion of these tools and their use.
- **Make SDF Files**—allows the developer to convert Access MDB files to the SDF format required by handhelds. Developers can ask the tool to create SDF files from the MDB files for Survey 0, Survey 1, or the CMSDB file, or all three by choosing different submenu options. The submenu choices and confirmation message are shown in the next screenshot. If the user chooses one of these options, the system automatically makes an SDF format copy of the designated MDB file. The file is placed in the location specified in the Setup Parameters Screen (select Program options from the main menu) at the SD Card Image path, in this example the root of the C: drive (C:\).



- **Help**—allows users to bring up the following GATS manuals or files:
  - ***GSS Guide***
  - ***Field Interviewer Manual***
  - ***Field Supervisor Manual***
  - ***Data Management Implementation Plan***
  - ***List of Day Codes***

  It uses the default viewer for PDF or text files on the PC to view these objects.

Program Options—allows users to set or save input parameters for various GSS modules. Users can create and save setup paths, passwords, and other required parameters for programs in one location and save them to be passed into programs as needed. The Setup Parameters screen is shown below:



The home directory is the path to the main tree of the GATS programs. The default for this location is C:\GATS_Folders and should not be changed. The SD card image path is the path to where SDF files are stored when converted from MDBs. The next three locations are the paths to the three GATS MDB databases files (Survey 0, Survey 1, and the CMSDB). The next three locations detail where the case files are stored, the help files are stored, and where the aggregation executables are stored. The encryption password is the password used to encrypt the SDF databases on the handheld. This should not be changed from the default for GATS. Finally, the screen title is user-changeable text that will be displayed on the developer's tools startup screen.

5.2 Questionnaire Designer

The Questionnaire Designer is a development tool that helps GATS developers create, edit, and maintain GATS survey forms. It allows GATS developers to fill the metadata tables required by GATS forms from an easy-to-use interface. Its function, along with adding some utilities, is to make filling in the Access MDB tables easier for the GATS developer and also make creating the SDF files needed for the handheld more automatic. It also allows the GATS developer to set paths and passwords one time and store them. The Designer does not alter the underlying table structure or metadata. Users can still modify GSS tables directly from Microsoft Access or directly edit the SDF tables from Visual Studio.

The Designer supports:

- Modifications of question and answer texts, type of questions, skip patterns, and logical checks.
- Validation of user inputs where feasible.

- Limits inputs of certain fields when they are not supported by a screen type; for example, ranges cannot be entered for list type questions.
- Selection of a second language from *n* languages to display and work in at one time. English will always be Language 0 and you can choose among the other languages defined in the databases for display as the second language.
- Automated version control for questionnaire updates—every time the questionnaire Designer saves a questionnaire program, it will update the programs version number (by convention the questionnaire program version number is stored in the text field for the START question). The format will be Ver. YYYY MM DD nnn, where YYYY is current year, MM is current month, DD is current day, and nnn is a sequential number that starts at 001 and is incremented by one with every save. A sample version number is as follows: Ver.: 2009 06 30 012.
- Import from Excel of selected questionnaire design components into questionnaire specs (i.e., importing question texts or answer texts in any language). This allows bulk import of translations or text from easy-to-use documents.
- Ability to save to MDB format (the default format for Designer files).
- Ability to save to SDF format for use on handhelds.
- Setup parameters that allow specification of paths and passwords for multiple use.
- Examples and definitions for the free text blocks for Special instructions of GSS Logic Function.
- A preview capability that shows a reasonable approximation of an iPAQ screen display, on a PC, for a given question. This feature provides only an approximation of the "true" iPAQ screen display but is helpful in getting a first approximation of how a given iPAQ screen is laid out and appears. This option is invoked by a Designer menu option called IPAQ Preview. It shows the questionnaire text in a PC view of the iPAQ screen for both languages currently in use.

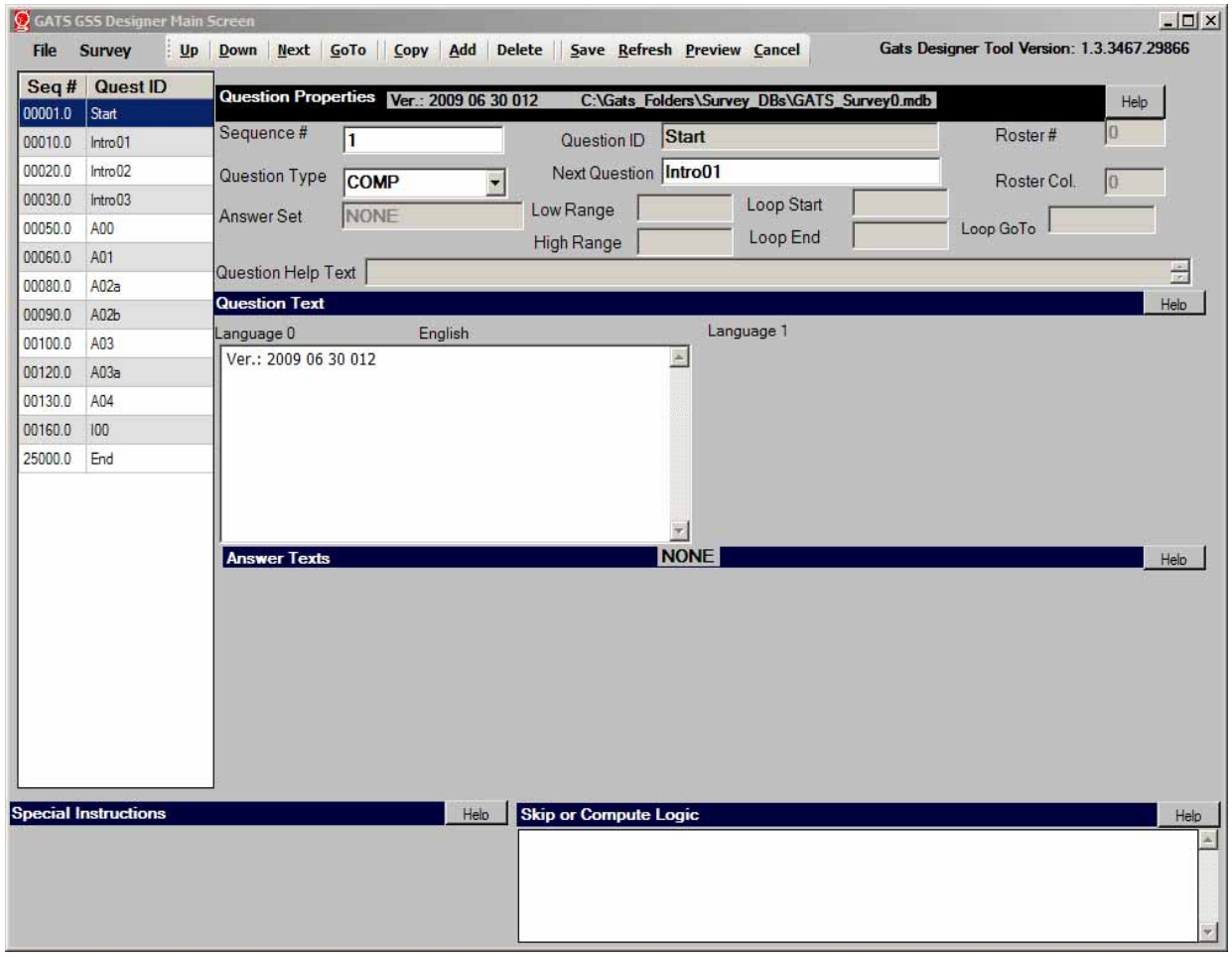
Exhibit 5-3 presents a screenshot of the Designer main screen.

Screen Blocks

The main blocks in the Designer screen are:

- **The QID list**—provides the list of the screens that make up a given survey. It is the leftmost area of the screen and shows sequence # and a QID name. Users can place the cursor on this list and use the keyboard navigation keys (up, down, page up, page down) to move about the list of QIDs.

Exhibit 5-3. Designer Main Screen



- **Question Properties Block**—contains items that pertain to each question (see **Chapter 4** for full description of these variables).
  - Sequence #
  - Question ID (QID)
  - Question Type
  - Answer Set Name if question type is ATA or List
  - Next Question
  - Range Data (Low and High) if question type is numeric
  - Loop controls if question type is Loop or Lend
  - Roster # and Roster Col
  - QxQ text
- **Question Text Block**—contains text for questions that display text. Users can work with up to two languages at a time. Language 0 is always English and Language 1 can be any one of the languages a user has provided in their Texts tables.
- **Answer Text Block**—provides the answer set lists if the question type is List or ATA for Language 0 and Language 1.

- **Special Instructions**—allows users to code special format instructions for a question (see **Chapter 4** for full description of this section).
- **Skip or Compute Logic Blocks (Qlogic)—**allows users to enter their Qlogic code. See **Chapter 4** and **Appendix A** for a discussion of the Qlogic syntax and options.

Help

Each section of the Designer has block-specific help that can be invoked. For example, the help for the Question properties is shown below.

> ***Question Properties Block***
>
> This section deals with the general properties of a question. The header of this block shows you the questionnaire version number and then the path to the MDB file that you are editing. The other items in this block are described in the table below.

Item	Description
Sequence # or Seq #:	A number (it can contain decimals) to control display or sort order of questions, for example, 10.2 or 1235. This number affects only the order in which questions are listed in the display grid; it does not affect the logical order of question execution.
Question ID or QID:	The name or ID of a given question. The name must be unique within a survey and >= 1 character and <= 20 characters.
Roster #:	The specific roster (can be 1 through 4), the roster is an internal grid for data storage, where you want a copy of the answer to the question stored.
Roster column:	The column of the roster where you want a copy of the answer to the question stored, the row is determined by the nesting or loop level.
Question Type:	This box allows you to select a type of question from the drop-down list of available question types (e.g., LIST or TEXT or NUM).
Next Question:	The name of the question ID you want to go to next by default (i.e., if there is no program logic that changes the logical flow; must be unique within a survey >= 1 and <= 20 characters).
Question Help:	Help text for the question that Field Interviewers can display during the survey when they tap the QxQ menu button in GSS on the iPAQ. At this time it can be in only one language. If there is no QxQ text then the QxQ option in the GSS questionnaire screen is greyed out.
Low Range:	Starting value of acceptable range for a numeric type question (e.g., 100.5).
High Range:	Ending value of acceptable range for a numeric type question, e.g., 200.8; for example, if the Low Range is 100.5 and the Hi Range is 200.8, then the numeric question will only accept numbers in the range 100.5 <= x <= 200.8.
Loop Start:	Starting variable of loop counter.
Loop End:	End value of loop.
Loop Goto:	Target QID to goto when loop is not yet finished (i.e., loop current counter < loop end).

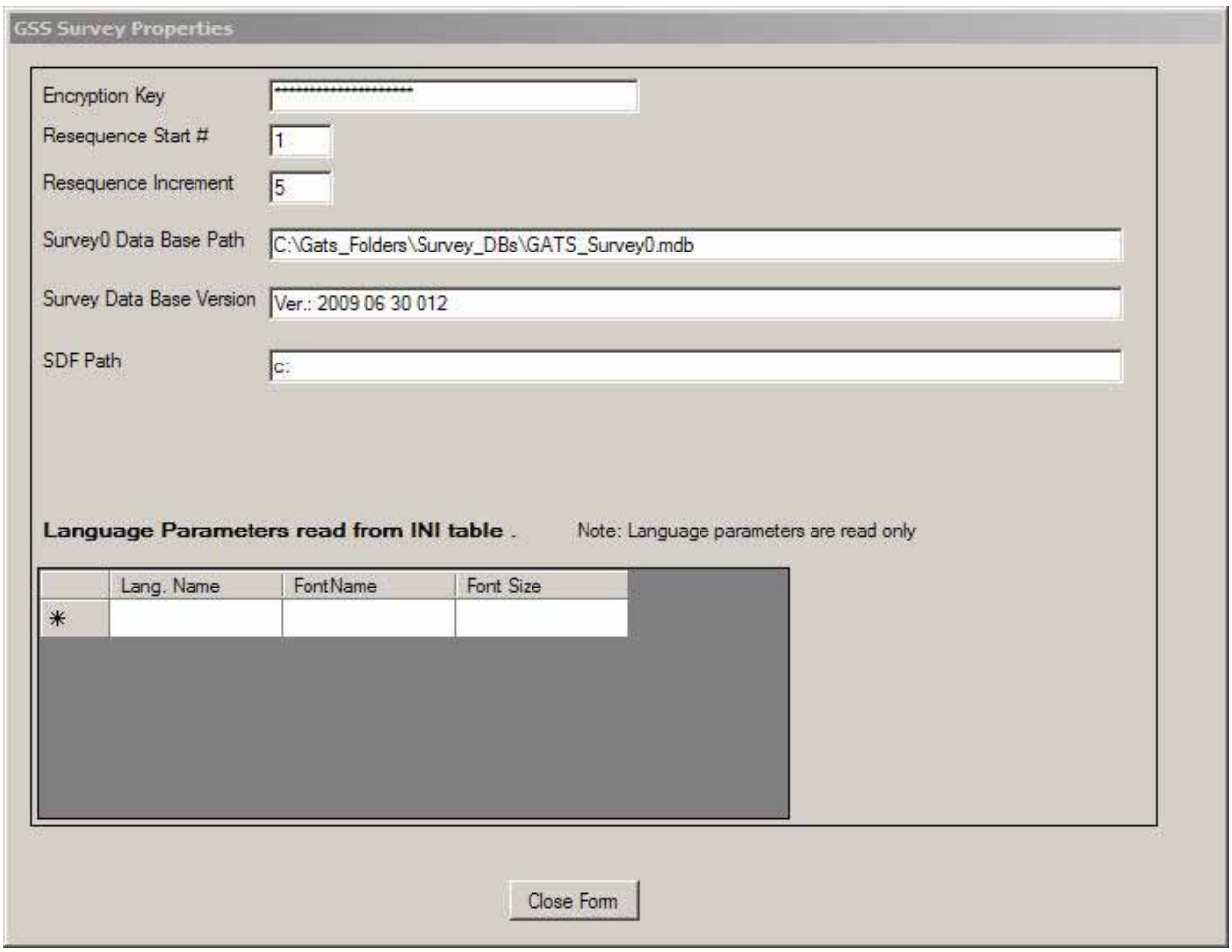
Menu Items

File

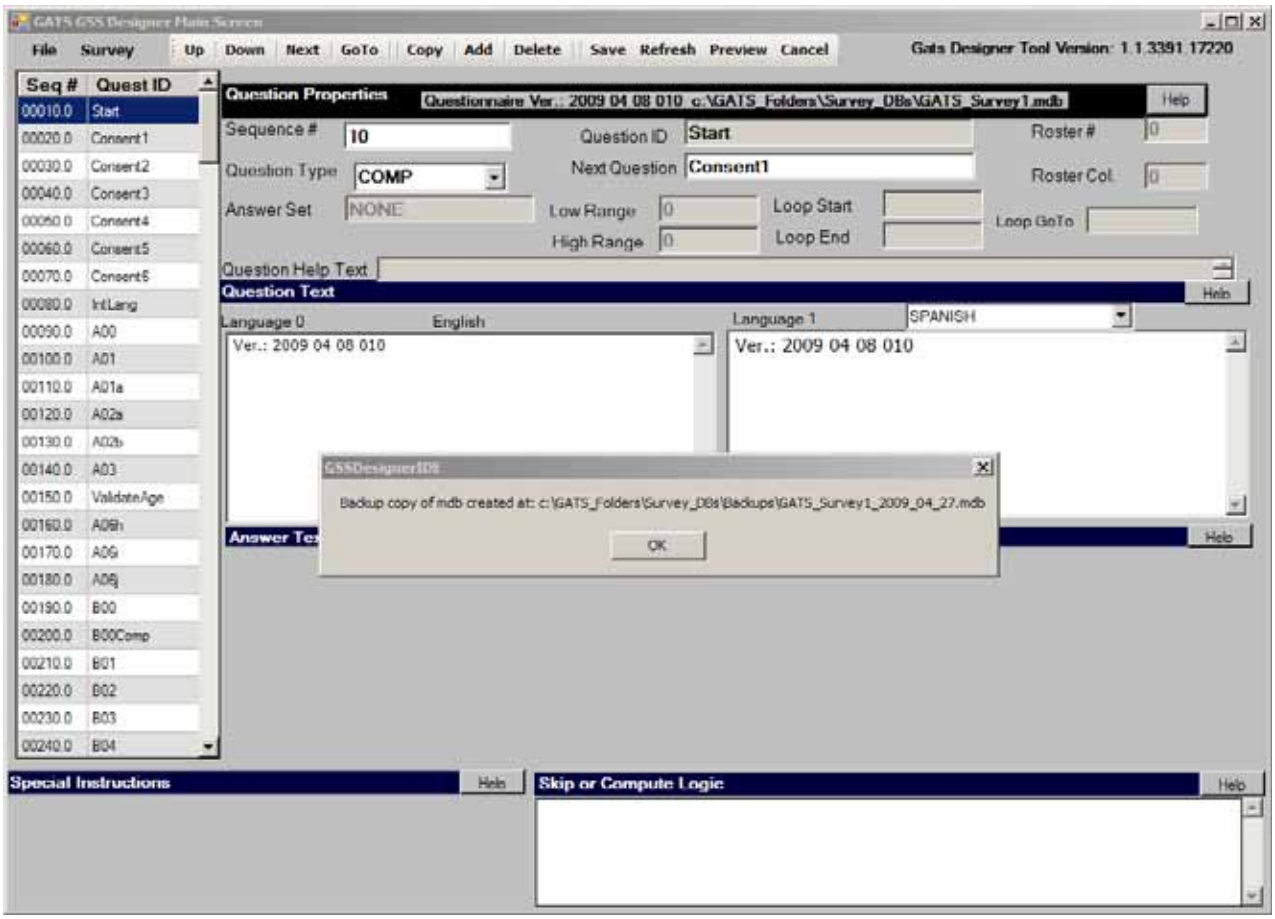
Under File, the options include:

- Save as SDF
- Import Data (Answers Text, Question Text, or QxQ Text)
- Make Backup
- Exit

The Save as SDF option creates an SDF version of the current MDB file in the target directory provided in the setup Survey Properties files SDF Path. The Designer allows users to create and set up paths in a properties screen. This is accessed from the Survey Menu and the properties submenu. The Designer Survey Properties screen is shown below:



The Make Backup option creates a backup of the current MDB. The naming convention and storage location of the backup are shown in the information box in the screenshot below:



The Import option allows importing bulk text for insertion into the Questions, Answers, or QxQ tables. This option could be used early in the development process when text is brought in from a translator. The import files must be in a tab-delimited format specific to the kind of table being imported. **Exhibit 5-4** shows the fields and order that must be used in the tab-delimited files for this option.

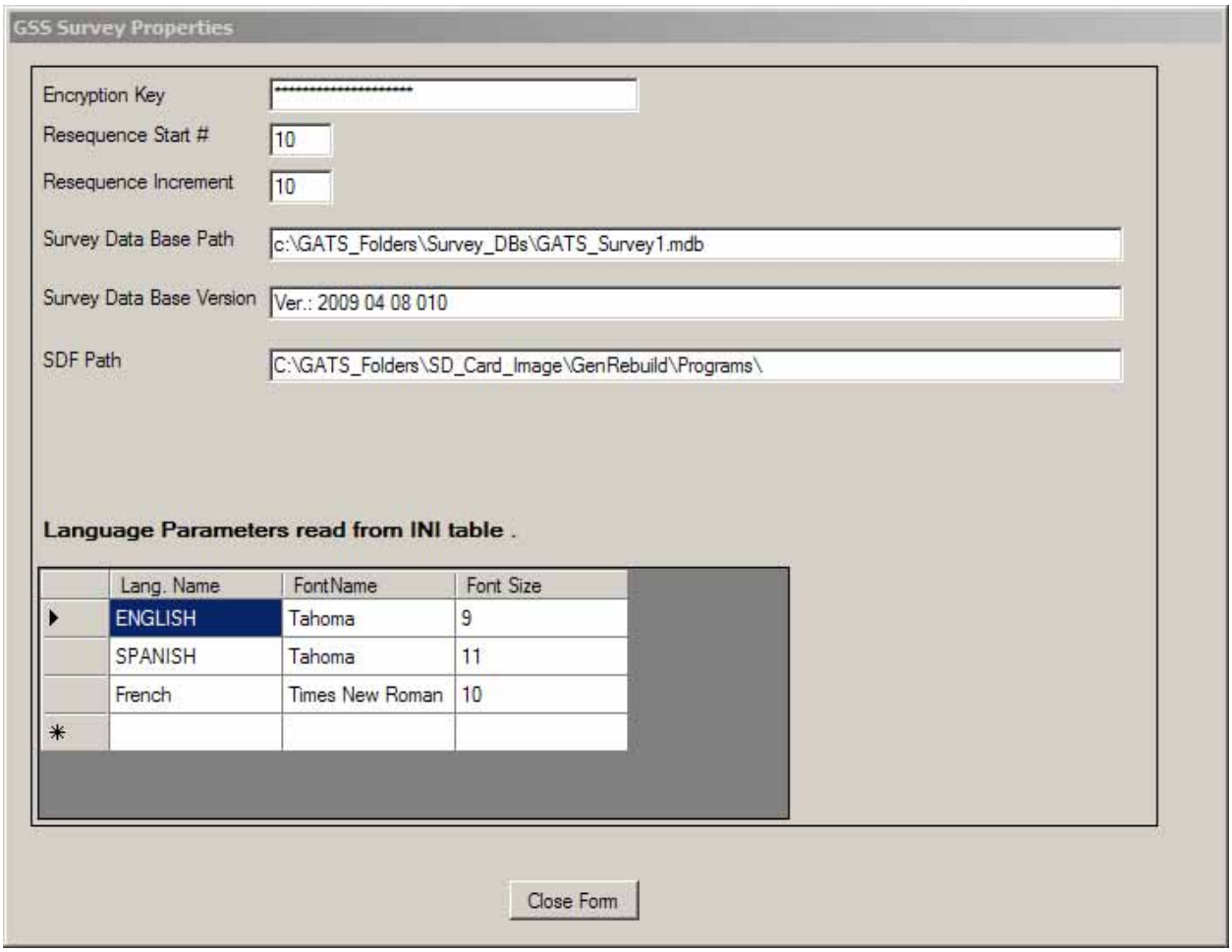
Exhibit 5-4. Import Layouts

Table Answer Cols	Table QxQ Cols	Table Questions Cols
AID	QID	QID
Language	Text	Language
Asequence		Text
Acode		
Atext		

Survey

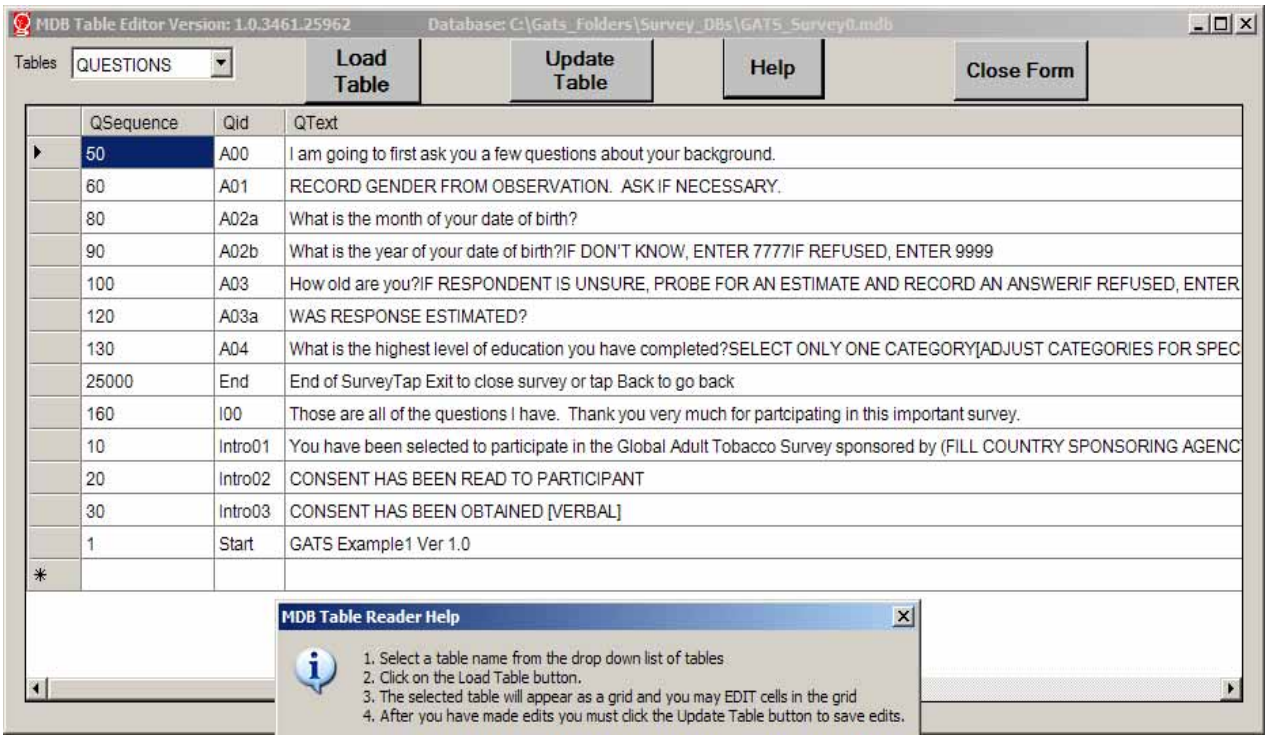
The Survey menu suboptions allow users to select subtasks that affect the entire survey. These are:

- **Resequence:** Causes the sequence numbers (Seq #) in the Questions table to be resequenced using the starting value and increment values that are in properties setup.
- **Properties:** Edit and save the properties data.



- **Update Version #:** Update the questionnaire Version number stored in the text for the Start QID. The version number will be updated with today's date and incremented by one from the previous version. For example, if the current version number is Ver.: 2009 07 06 010 and today is July 6, 2009 then the next or updated version # would be Ver.: 2009 07 06 011.

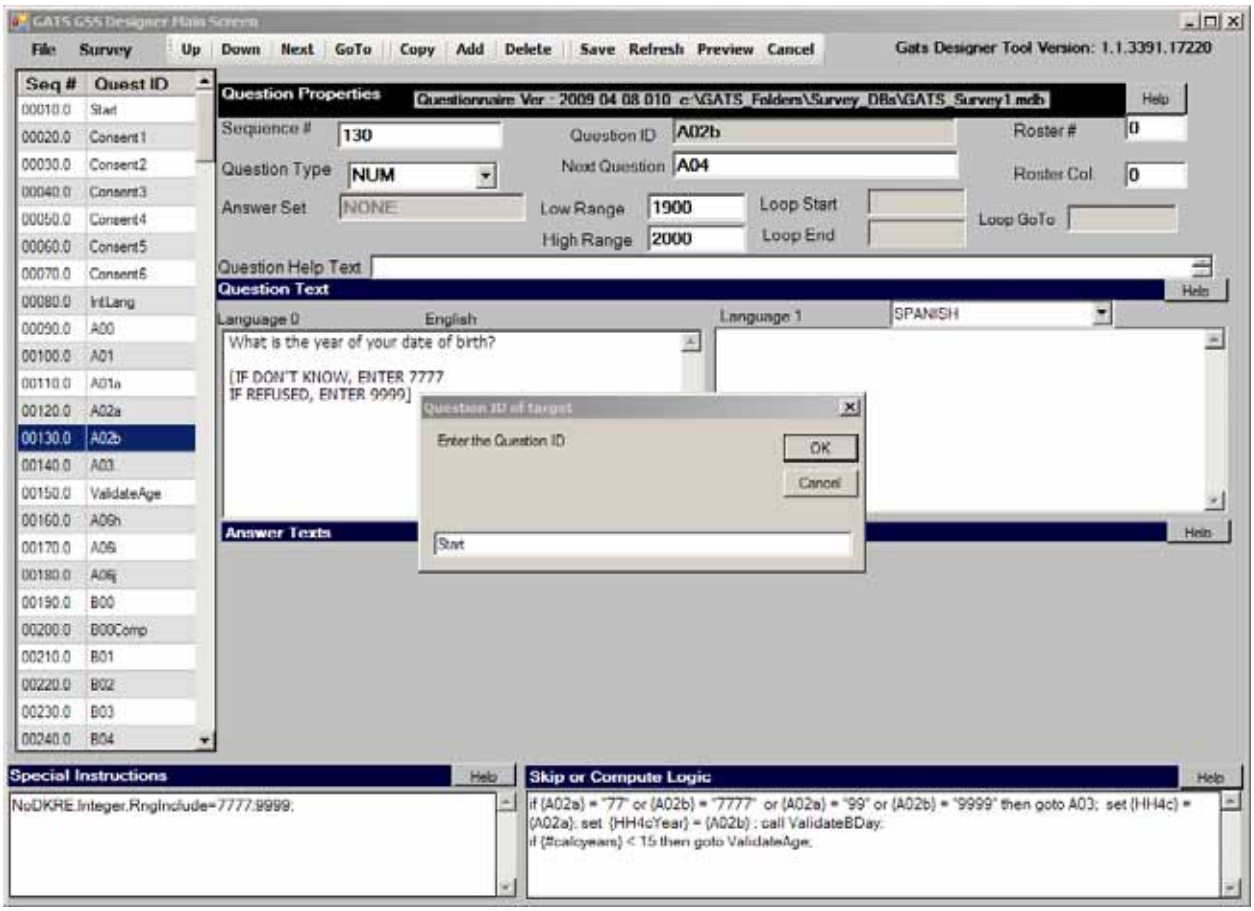
- **Edit Database Tables:** Edit or view the underlying database tables for Survey0, Survey1, or the CMSDB; this allows the developer to edit INI tables of other tables not directly accessed by the IDE. This function is also available in the Main Menu program. Here developers can directly review and edit internal tables in the Survey0 and 1 (Tables: Answers, INI, Questions, Texts, QxQ) databases and the CMSDB database (Tables: EvtDef, INI, Messages, TrainingCases). The TrainingCases and Answers table cannot have new rows added. The screenshot of this option below shows the Questions table open for Survey0 and the help file open as well. The user can edit the grids for all tables except the Answers table and add new rows. See the help screen in the next screenshot that explains how to operate the edit options.



The other main menu items of the IDE fall into a few groups: Navigation (Up, Down, Next, GoTo), Management (Add, Delete, Copy), Admin (Save, Refresh, Cancel), and View (Preview). These controls are described below.

Navigation Menu Items

- **Up/Down**—allows you to move up or down in the list of QIDs (you can also use the up and down arrows or click on a given QID).
- **Next**—goes to the QID specified in the GSS Next Question Block.
- **GoTo**—prompts you for a specific QID, as shown in the next screenshot.



Admin Menu Items

- **Copy/Add/Delete**—allows you to:
  - add a new QID screen with a new QID,
  - copy an existing QID screen and rename it, or
  - delete an existing QID screen.
- **Save**—forces an immediate save of the current screen that is displayed. Normally a screen is saved when you move from it to another QID screen.
- **Refresh**—causes the current QID screen to be refreshed by reading in the stored data for it.
- **Cancel**—ignores the current changes to a QID screen and brings back the QID screen contents as of the last Save. Cancel allows you to cancel any changes made to a screen before you move. It retrieves the saved data from the last time the screen was saved. This allows you to cancel changes you may have made in error or no longer like before moving from the screen, thereby causing the screen to be saved.

View Menu Items

Preview—displays a preview screen that shows an approximation of what the iPAQ screen will look like given the current parameters for the QID. A sample preview of a Num type screen is shown in the next screenshot with both English and Mandarin text.

5.3 Data Aggregation Software

View Data—Data Viewer

Overview

The RawDataViewer is a basic tool for inspecting SDF data. It offers the following options:

- View CaseIDs
- View Field Interviewer IDs
- View events data
- View responses data
- View tabulations of the question responses
- Export the data to CSV

Terminology

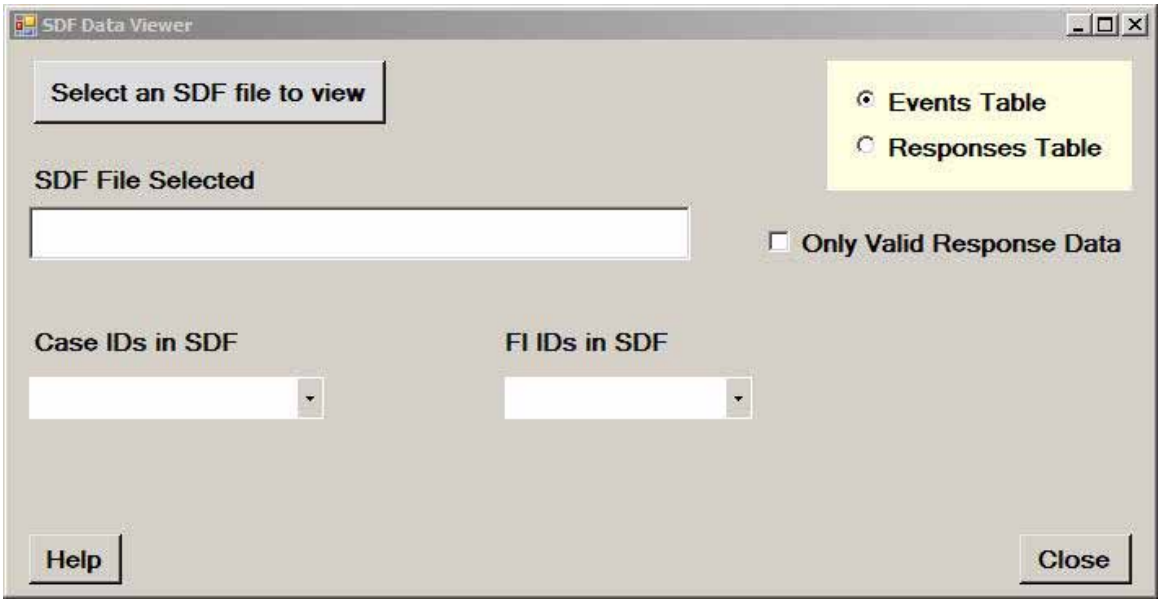
- Field Interviewer: Field Interviewer
- SDF: Small Device Format

Process

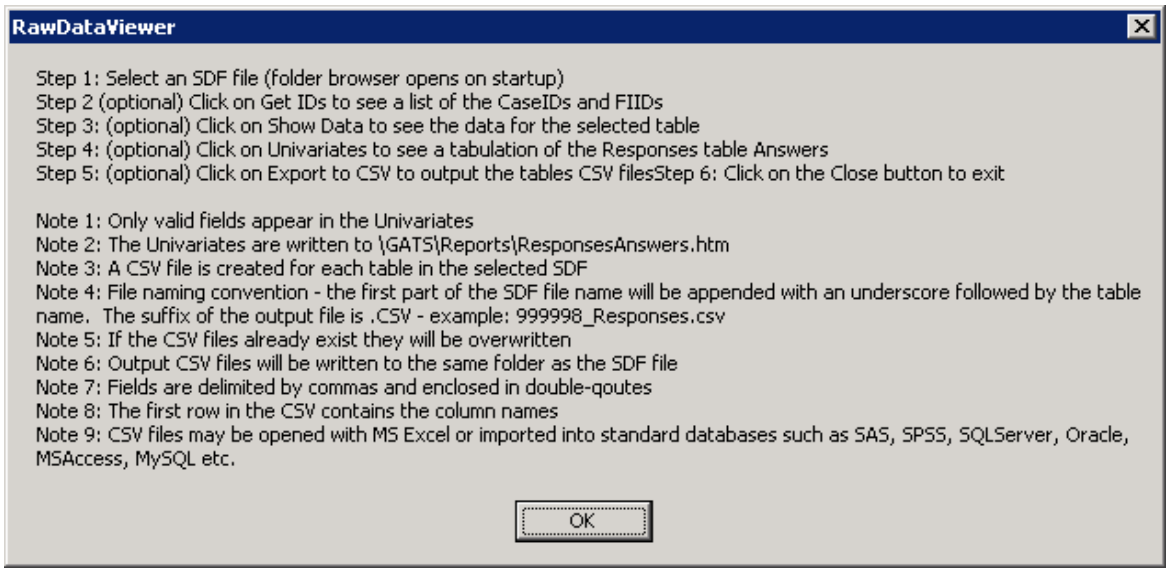
The RawDataViewer tool is to be used on an ad-hoc basis to inspect data.

How to Use the RawDataViewer

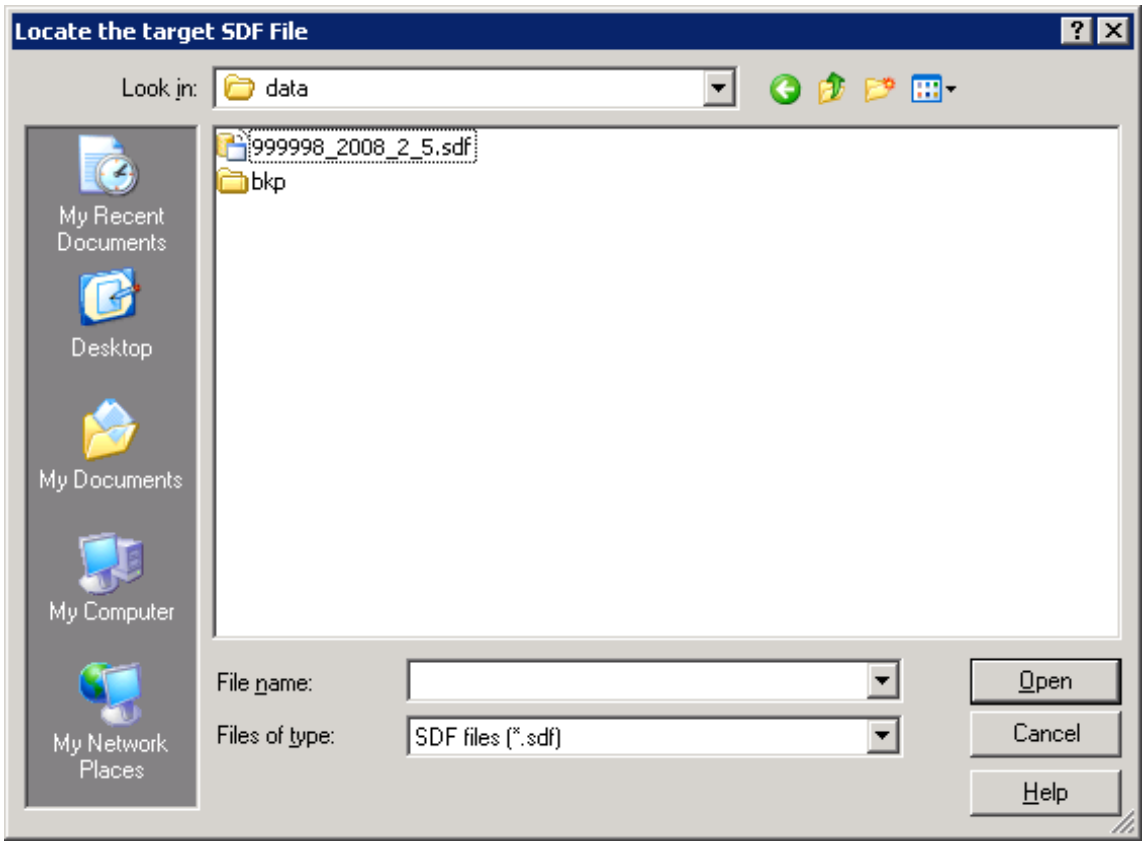
1. In the GSS Developer Tool set, invoke the RawDataViwer from the Data Aggregation Menu sub-item View Data.



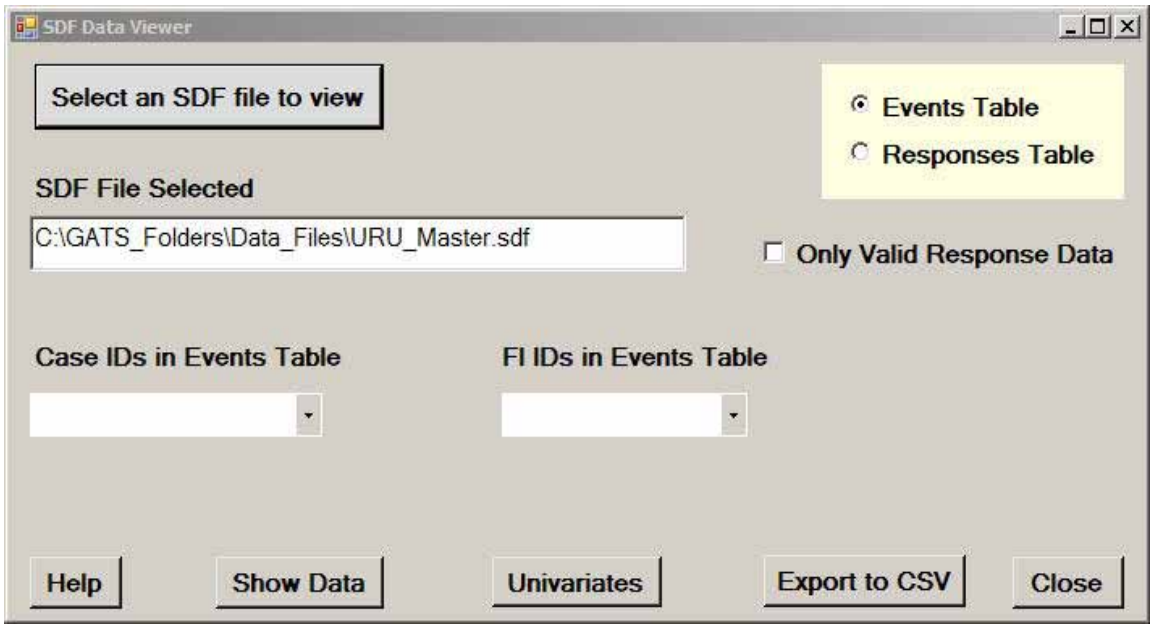
2. Click Help for detailed instructions and notes.



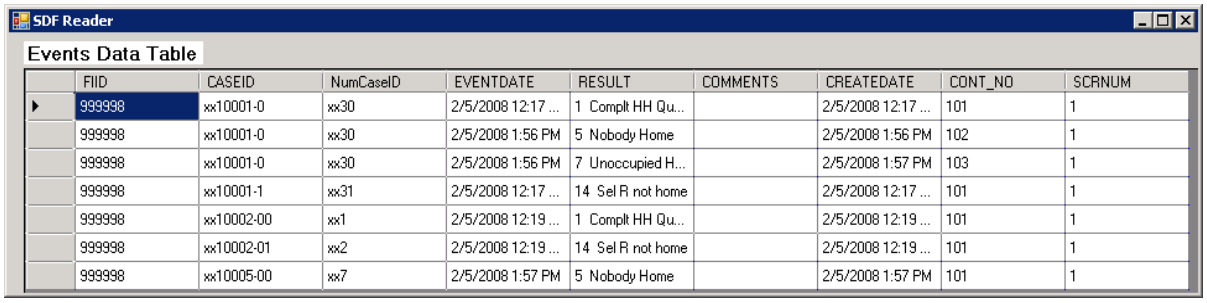
3. Click Select a data source directory.



4. Click Open.

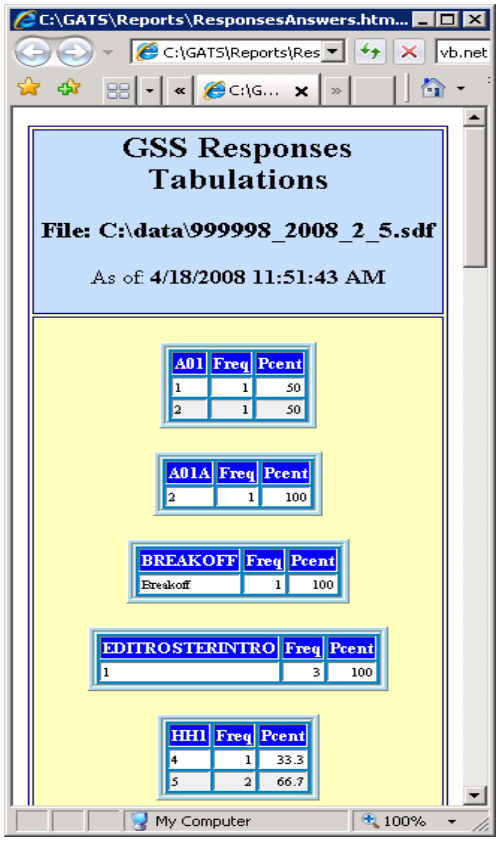


5. Click the **Show Data** button.
   The **Show Data** button displays the data from the responses or events table, as selected. Users can also select a specific Case ID or Field Interviewer ID to filter the data view.



FIID	CASEID	NumCaseID	EVENTDATE	RESULT	COMMENTS	CREATEDATE	CONT_NO	SCRNUM
999998	xx10001-0	xx30	2/5/2008 12:17 ...	1 Complt HH Qu...		2/5/2008 12:17 ...	101	1
999998	xx10001-0	xx30	2/5/2008 1:56 PM	5 Nobody Home		2/5/2008 1:56 PM	102	1
999998	xx10001-0	xx30	2/5/2008 1:56 PM	7 Unoccupied H...		2/5/2008 1:57 PM	103	1
999998	xx10001-1	xx31	2/5/2008 12:17 ...	14 Sel R not home		2/5/2008 12:17 ...	101	1
999998	xx10002-00	xx1	2/5/2008 12:19 ...	1 Complt HH Qu...		2/5/2008 12:19 ...	101	1
999998	xx10002-01	xx2	2/5/2008 12:19 ...	14 Sel R not home		2/5/2008 12:19 ...	101	1
999998	xx10005-00	xx7	2/5/2008 1:57 PM	5 Nobody Home		2/5/2008 1:57 PM	101	1

6. Click the **Univariates** button.
   This option produces frequency reports for all QIDs that appear in the responses table.



7. Click the **Export to CSV** button.
   This option creates CSV files for the following tables in the selected SDF file:
   - Responses
   - Notes
   - INI
   - DUEVT
   - DU
   - AggLog
   - Addresslog

8. Click the **Close** button.

Configuration

Hardware:

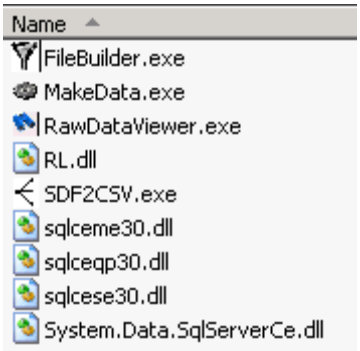
- Laptop with 1 GB RAM and 40 GB of free disk space

File structure:

C:\

Software:

C:\GATS\bin



5.4 Aggregate Data—File Builder Tool

Overview

File Builder is a program that combines or aggregates Field Interviewer iPAQ data files. File Builder should be used for aggregating data from the individual iPAQ through as many field levels as are

implemented for a given country. The File Builder program facilitates collecting and combining the iPAQ files with the intermediate aggregated files into one cumulative master data file residing on a PC.

File Builder works on SQL Server Compact Edition files (SQL CE SDF), which are supported on both the iPAQ and Windows PC platforms. The GSS software can export SDF files, capturing all the internal data files on the iPAQ. These exported files can then be aggregated together at various data collection points within a country to build a master data file that contains all the questionnaire response data, event data, notes, dwelling unit, and address changes for the survey.

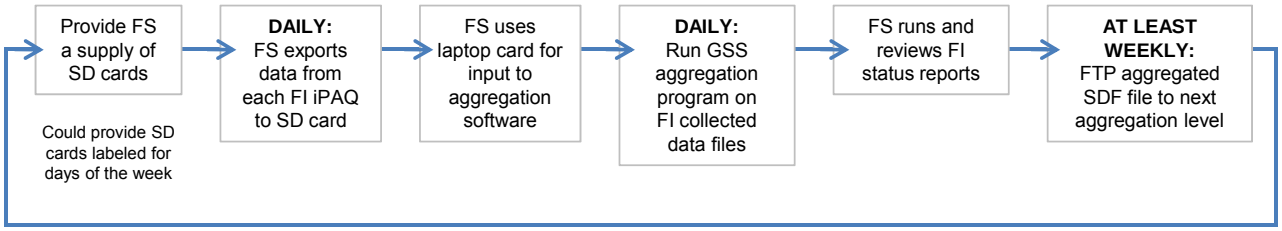
This section covers the use of the File Builder tool from application setup to closing, so that iPAQ data may be aggregated across all levels. Specifically, this section addresses the following:

1. Using File Builder
2. Setting up File Builder on a PC
3. Running File Builder
4. Output generated by File Builder
   a. Reports
   b. Master data file
5. Closing File Builder
6. Next steps

1. Using File Builder

Field Supervisors, Regional Supervisors, State Coordinators/Supervisors, and the National Supervisor will use File Builder to aggregate data from each field level below them. For example, a Team Leader with several Field Interviewers reporting to him/her could use File Builder to aggregate the data from each of his/her Field Interviewer's SD cards into the master data file on the Team Leader's laptop. The Team Leader's master data file will then be input to the Regional Supervisor's File Builder process. The Regional Supervisor's master data file will then be input to the State Supervisor's master data file. Finally, the State Supervisor's master data file will be input to the National Data Center's File Builder process. In the end, all data from all sources will be aggregated into the National Data Center's master data file using File Builder. **Exhibit 5-5** illustrates the general process involved in data aggregation at the lowest level.

Exhibit 5-5. Aggregation Diagram



Note: FS is a Field Supervisor, FI is a Field Interviewer

2. Setting Up File Builder on a PC

For most countries, Field Supervisors, Team Leaders, State Coordinators, and the National Data Center are expected to use a PC for data aggregation. The File Builder tool is for use on a PC/laptop and requires a one-time installation.

The following PC hardware is the minimum required to install and run File Builder:

- A laptop with 1 GB RAM and 40 GB of free disk space
- An SD card reader
- One SD card

The setup steps for the PC are:

- **Step 1:** Run the supplied installation utility.
- **Step 2:** Ensure that the SDF data to be aggregated are readily accessible and reside on the PC.
- **Step 3:** Update a machine identification file, called a node file, on the PC.

Copying File Builder to the Target PC

For use at the PC level, you must copy the File Builder software to the C:\GATS\ folder so that the folder structure, files, and executable code exist. The software is supplied by an install utility. The IT/computer specialist who sets up the PCs for field use should use the utility to install the software. It will result in the following set of folders:

SDF Field Interviewer Files to be Aggregated

The data files to be aggregated must exist as SDF files in an accessible location, such as on the C:\GATS\Incoming (you should create a folder in like this to store your inbound sdf files) folder of the PC running the File Builder tool. Data must be exported from the iPAQs to the PC ahead of the File Builder data process execution.

The iPAQ SDF data file transfer (or export) can be performed as described in the following steps:

1. Field Interviewer to next level (e.g., Field Supervisor/Team Leader) aggregation:
   - Field Supervisor removes the operational SD card from the Field Interviewer's iPAQ.
   - Field Supervisor inserts the aggregation SD card into the Field Interviewer's iPAQ.
   - Field Supervisor runs Export to copy data from the iPAQ to the aggregation SD card.
   - Field Supervisor removes the aggregation SD card from the Field Interviewer's iPAQ.
   - Field Supervisor re-inserts the Field Interviewer's operational SD card into the Field Interviewer's iPAQ.
   - Field Supervisor inserts the aggregation SD card into the laptop SD card reader.
   - Field Supervisor runs File Builder.
2. All other aggregation:
   - Assumption: SDF data have been transferred from lower levels to the current node (FTP, diskette, CD, SD card, memory stick, etc.) to a specific folder on the node (e.g., C:\GATS\Incoming).
   - Node administrator runs File Builder.

For detailed instructions on using the iPAQ GSS Export utility, see the ***GATS Field Interviewer Manual***, Section 9C.1.

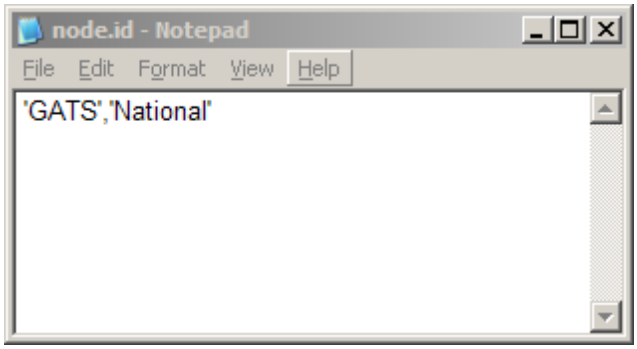
Creating a Node File

A node file is a necessary piece of information you set that identifies the aggregation level/person/machine and is key to the aggregation building process. In the C:\GATS folder, a file called node.id must be edited using a Windows editor such as Notepad, Textpad, or any other text-file editor. The file is to contain the information used to index the aggregation log entries. The node.id file must contain a unique name for each node ID, or the data will not be moved up to the next aggregation level. Once data have been aggregated, the node.id file should not be modified.

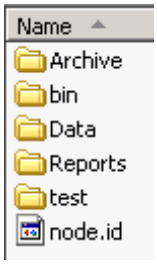
The node.id file must contain a single record with the information starting in column 1, row 1, specifying two pieces of information: the node level designation, such as the Field Supervisor, and the identifier/name of the node.

Single quotes and comma are required for all fields entered (e.g., “GATS” or “NDC”).

Example of a node file:



The Node File should be named **node.id** and be saved to the GATS folder.



You should now create a file in the C:\GATS\Data folder call Master.sdf. You do this by copying the file C:\GATS\Data\Master.sdfzero and renaming the copy to Master.sdf.

3. Running File Builder

To initiate File Builder you may invoke it from the GSS IDE Main Menu Data Aggregation menu item—Aggregate Data—or double-click the FileBuilder.exe file in the C:\GATS\bin folder.

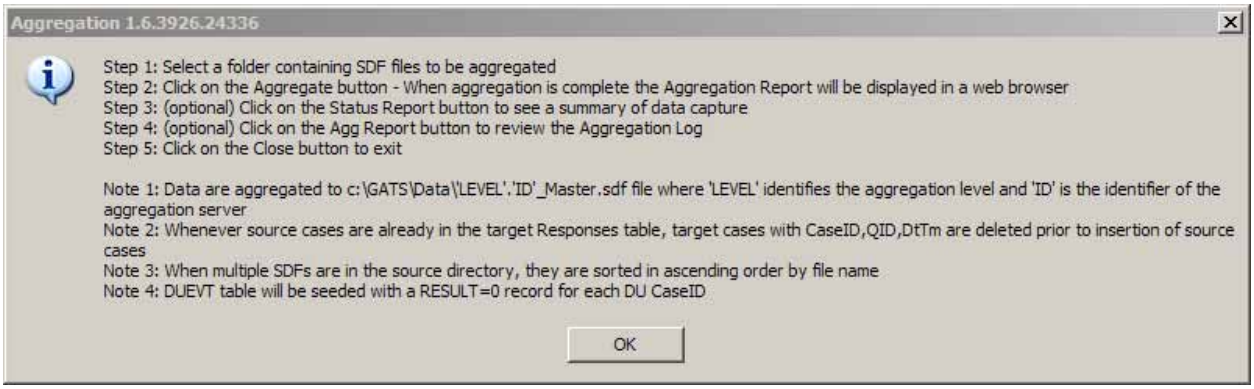
Note the following points about File Builder:

- File Builder combines the SDF files to be aggregated in **ascending order by filename**, hence newer files of the same name overwrite older ones.
- File Builder aggregates five tables from the input SDF files; these are
  - AddressLog (address change data)
  - DU (case information, address, sample info, etc.)
  - DUEVT (event data)
  - Notes (interview notes about the cases)
  - Responses (questionnaire data)
- File Builder creates a record-level "key" for the records in the input tables using the CaseID.
- File Builder does not allow for duplicate records in the aggregated file, based on the CaseID.
- File Builder creates a single, aggregated master SDF file and does not alter the input SDF input files. Ideally, the input folder should contain the most recent files that need to be aggregated. A separate folder may be created that contains the SDF files that have already been aggregated, so they are not aggregated again. Please note that exported SDF files from an iPAQ are always a complete copy of the data so that information that was exported in earlier files will also be present.
- The master SDF file always resides in the GATS\Data folder and inbound aggregation files should NOT be in this folder.
- Every step in the aggregation process is cumulative.

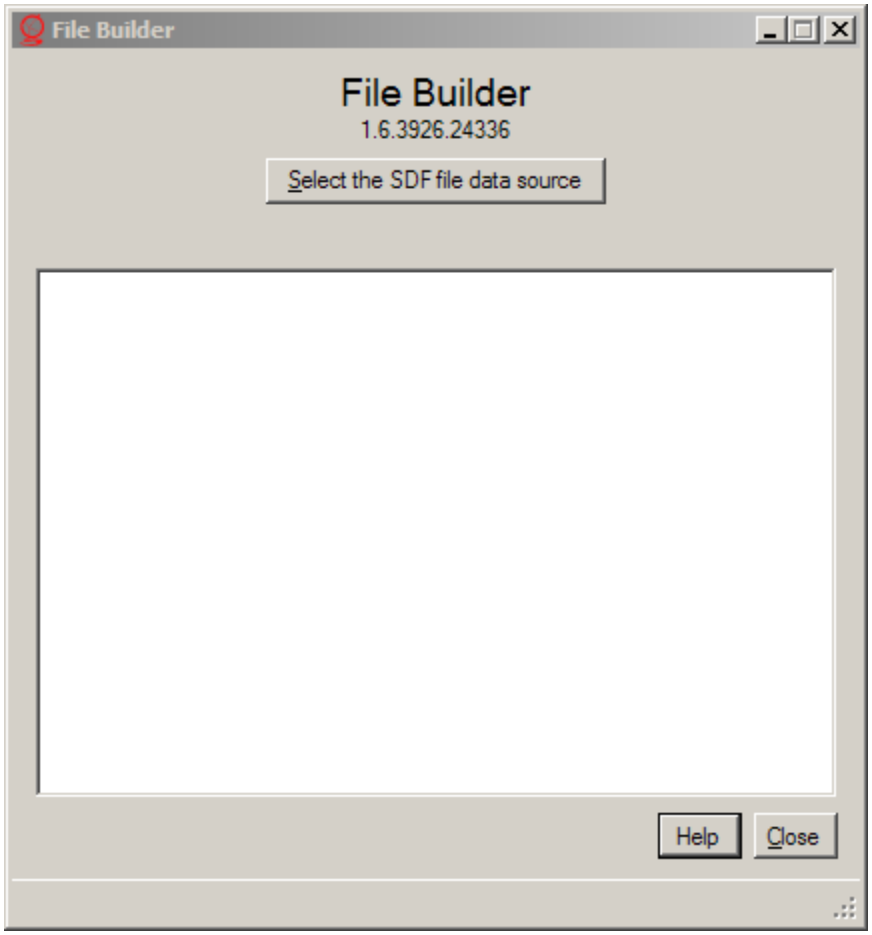
At this point in the process, the following screen displays, indicating that the user should enter a single data source directory in which all the data files to be aggregated reside.

File Builder Help

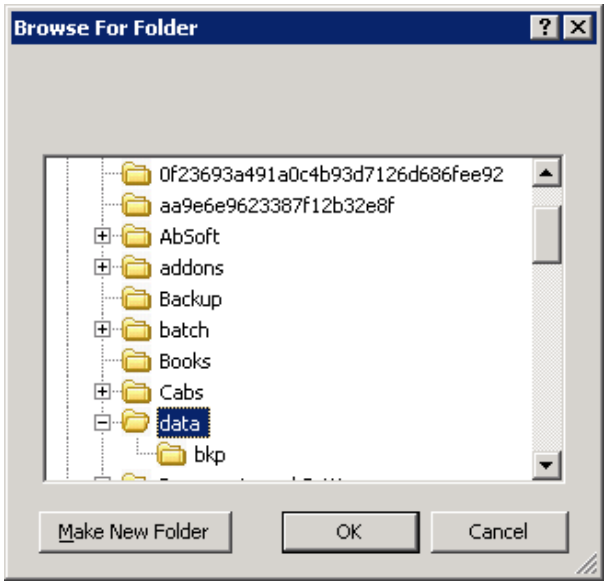
A Help screen, available within File Builder, provides some instructions and notes. To access the Help screen, click the **Help** button, and the following screen appears.



Click on **Select the SDF file data source**, and browse for the directory/folder in which the iPAQ data were previously saved. While it is not necessary to maintain the iPAQ data files in a certain folder, they **should not** be stored in C:\GATS\Data folder on the PC where the master aggregated file will be kept.

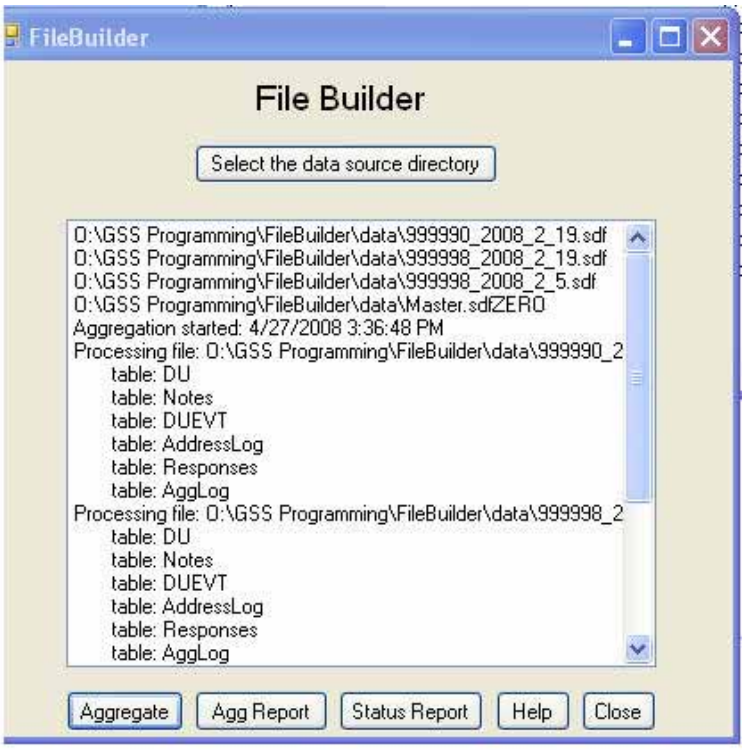


Once the directory containing the SDF data has been identified, click on the folder name, and then click **OK**.



The SDF files residing in the PC folder that you selected are then listed on the File Builder screen.

Click the **Aggregate** button. Doing so will aggregate/combine all the files in the folder. A screen is displayed indicating the tables that have been read in from the aggregated SDF files.



4. Output Generated by File Builder

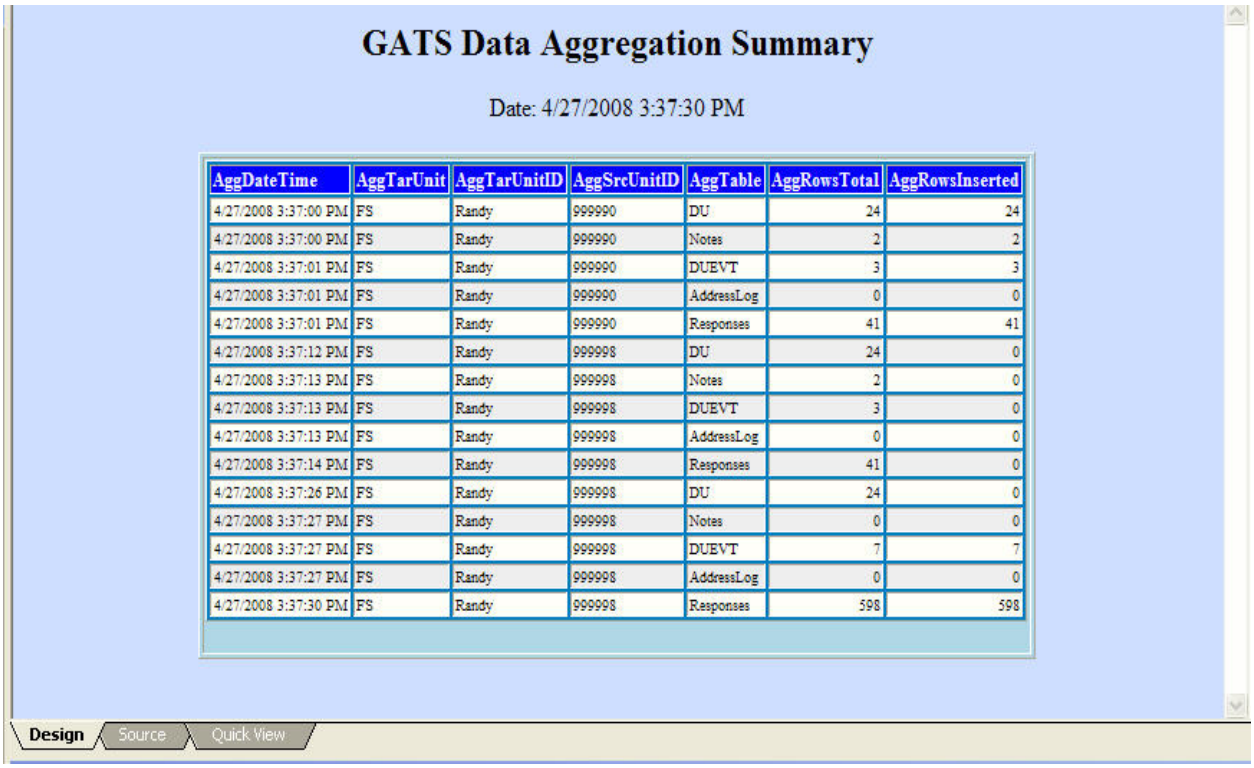
a. Reports

Once the data have been aggregated successfully, File Builder can provide several reports, detailed below.

Aggregation Summary Report—directly after the aggregation summary screen is displayed and the aggregation is complete, a summary table showing the log of the aggregation is generated. The contents of the generated file aggrpt.htm are displayed using the default Windows HTML editor on the PC. See **Exhibit 5-6** for a screenshot of the Aggrpt file.

The aggregation summary report can be generated by clicking the **Agg Report** button that is displayed after files have been successfully aggregated.

Exhibit 5-6. Aggregation Summary Report (file Aggrpt.htm)



GATS Data Aggregation Summary

Date: 4/27/2008 3:37:30 PM

AggDateTime	AggTarUnit	AggTarUnitID	AggSrcUnitID	AggTable	AggRowsTotal	AggRowsInserted
4/27/2008 3:37:00 PM	FS	Randy	999990	DU	24	24
4/27/2008 3:37:00 PM	FS	Randy	999990	Notes	2	2
4/27/2008 3:37:01 PM	FS	Randy	999990	DUEVT	3	3
4/27/2008 3:37:01 PM	FS	Randy	999990	AddressLog	0	0
4/27/2008 3:37:01 PM	FS	Randy	999990	Responses	41	41
4/27/2008 3:37:12 PM	FS	Randy	999998	DU	24	0
4/27/2008 3:37:13 PM	FS	Randy	999998	Notes	2	0
4/27/2008 3:37:13 PM	FS	Randy	999998	DUEVT	3	0
4/27/2008 3:37:13 PM	FS	Randy	999998	AddressLog	0	0
4/27/2008 3:37:14 PM	FS	Randy	999998	Responses	41	0
4/27/2008 3:37:26 PM	FS	Randy	999998	DU	24	0
4/27/2008 3:37:27 PM	FS	Randy	999998	Notes	0	0
4/27/2008 3:37:27 PM	FS	Randy	999998	DUEVT	7	7
4/27/2008 3:37:27 PM	FS	Randy	999998	AddressLog	0	0
4/27/2008 3:37:30 PM	FS	Randy	999998	Responses	598	598

Design Source Quick View

Status Report—click the **Status Report** button on the File Builder aggregation screen to generate a status report summarizing the data from the SDF files will appear (file statrpt.htm). See **Exhibit 5-7** for a screenshot of the statrpt.htm report.

You can generate the status report by clicking the **Status Report** button that is displayed after files have been successfully aggregated.

This Data Aggregation Summary Report provides information about the results of the aggregation. Each row shows the results for an aggregation unit (typically a Field Interviewer) and a specific database table.

The table is denoted in the AggTable column, the number of rows in the input table is shown in the AggRowsTotal column, and the number of rows inserted from that table is shown in the last column AggRowsInserted. The report above shows in the first line that for Field Interviewer 999990 and the DU table, there were 24 rows and all 24 rows were inserted into the master aggregation file

The Household Screening Status report is a summary of the status of all cases in the aggregated data file that were reported. For each aggregation unit (typically a Field Interviewer) the total number of cases is reported for that unit in the Cases column. Then the total is broken out into the various status categories. The status columns (Completed Ints, Unworked, Pending, and Final) depend upon the most recent result code for the case. The result codes are defined in the INI table of the CMSDB data base. The typical GATS values are shown below:

INI VariableName	VariableValue	Comment
EvtcodeComplete	200, 400	Result codes that are complete quex
EvtCodeFinal	200, 201, 202, 203, 204, 205, 206, 208, 209, 999, 400, 401, 402, 403, 404, 407, 408, 409	Result codes that are final status codes
EvtCodePending	0, 102, 103, 104, 105, 106, 108, 109, 302, 303, 304, 307, 308, 309, 887	Result codes that are pending status codes
EvtCodeScreened	200	Result code indicating screening complete and one person selected for more data collection

Unworked cases have a most recent result code of 0. Note that the final status includes the cases that are completed interviews. The cases column is the total for a given aggregation unit and should equal the sum of the Pending, Unworked, and Final columns. In the status report for the *Individual Questionnaire*, cases are filtered. Only households that have an result code 200 from household screening (screening interview complete and eligible HH member selected) are shown in this table.

Exhibit 5-7. Status Report (Statrpt.htm)

GATS Household Screening Status

Date: 10/4/2010 10:42:16

Unit	ID	Cases	Unworked	Pending	Final	CompletedInts	Date of Most Recent Data
FI	999998	7	0	0	7	5	6/30/2010 14:34:38
FI	999997	2	0	0	2	1	6/30/2010 13:44:51
FI	100000	3	0	0	3	2	6/30/2010 14:31:42
Totals		12	0	0	12	8	
%		100	0	0	100	66.7	

- Notes: "Cases" is the sum of the "Unworked", "Pending" and "Final" cases. "Pending" includes cases that have been worked, but not finalized (result codes 102,103,104,105,106,108,109). "Final" includes completed interviews and other finalized cases (result codes 200,201,202,203,204,205,206,208,209). "CompletedInts" includes result code 200.

GATS Individual Questionnaire Status

Date: 10/4/2010 10:42:16

Unit	ID	Cases	Unworked	Pending	Final	CompletedInts	Date of Most Recent Data
FI	999998	4	0	0	4	3	6/30/2010 14:49:24
FI	999997	1	0	0	1	1	6/30/2010 13:53:21
FI	100000	2	0	0	2	2	6/30/2010 14:56:00
Totals		7	0	0	7	6	
%		100	0	0	100	85.7	

- Notes: "Cases" is the sum of the "Unworked", "Pending" and "Final" cases. "Pending" includes cases that have been worked, but not finalized (302,303,304,307,308,309). "Final" includes completed interviews and other finalized cases (result codes 400,401,402,403,404,407,408,409). "CompletedInts" includes result code 400.

b. File Builder Output Data File—the Master Aggregated SDF

Once the aggregation has successfully completed, File Builder will create a single combined SDF data file, considered the master data aggregation file, located on the PC in the C:\GATS\Data folder. The first time File Builder is run, the master.sdf file is named using the node.id information (e.g., GATS_National_Master.sdf). The original SDF file(s) used in the aggregation will reside as it is in its previously saved location.

Browse the C:\GATS\Data folder to find the master SDF file that the aggregation process has created.

5. Closing File Builder

If you click the **Close** button at any time during the aggregation process, the File Builder application closes, and you will exit from the application.

Once the aggregation process has successfully completed and the aggregation and status reports have been generated, click the **Close** button to exit File Builder.

6. Next Steps

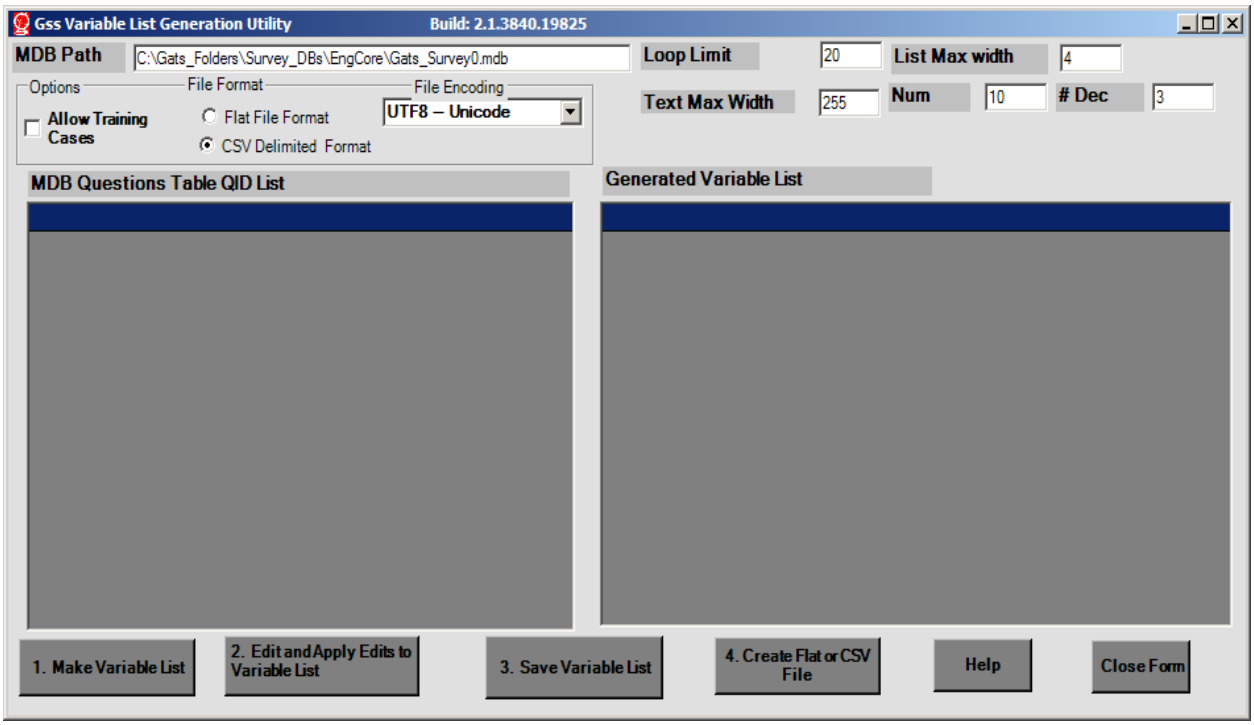
The next steps to be performed depend on the current level of aggregation. The master SDF file created by the aggregation process may now be sent to the next level (via FTP or other means) and aggregated there, as needed.

Transpose Data Utility

This utility provides the capability of taking the aggregated Responses data and creating a variable list and files for Survey0 and Survey1. It is a VB.net program that reads the MDB for a given survey and makes a variable list from this MDB. Using the variable list as a template, it extracts the response data from the aggregated SDF file and makes a single data record for each unique ID in the SDF file.

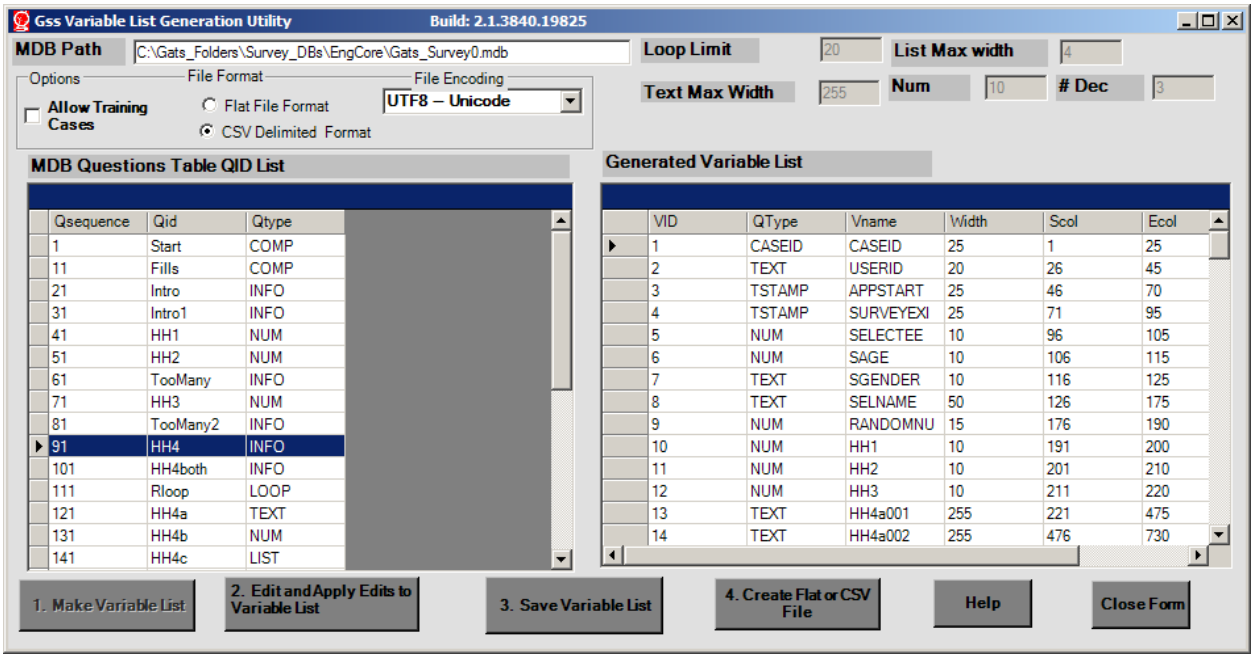
The opening screen for the transpose utility is shown in the following screenshot. You select the Survey0 or 1 by editing the MDB path block. Six data blocks on the screen allow users to enter variables that drive the construction of the variable list and flat file. The values shown in the screenshot are the suggested defaults. The definitions of the variables are as follows:

- Loop Limit—maximum number of “instances” of variable in a loop for the GATS *Household Questionnaire* 20 is used
- Text Max Width—maximum width, in characters, of a text type item
- List Max Width—maximum width, in characters, of a list type item
- Num—maximum width of numeric fields
- # Dec—number of decimal places



You can select whether or not to allow training cases by checking the Options box for Allow Training Cases. You can also select the format for the output file (flat file or CSV file) and its encoding. Again, the selected options in the figure are the recommended defaults.

To start the process, tap the **Make Variable List** button. The following screen displays:



Qsequence	Qid	Qtype
1	Start	COMP
11	Fills	COMP
21	Intro	INFO
31	Intro1	INFO
41	HH1	NUM
51	HH2	NUM
61	TooMany	INFO
71	HH3	NUM
81	TooMany2	INFO
91	HH4	INFO
101	HH4both	INFO
111	Rloop	LOOP
121	HH4a	TEXT
131	HH4b	NUM
141	HH4c	LIST

VID	QType	Vname	Width	Scol	Ecol
1	CASEID	CASEID	25	1	25
2	TEXT	USERID	20	26	45
3	TSTAMP	APPSTART	25	46	70
4	TSTAMP	SURVEYEXI	25	71	95
5	NUM	SELECTEE	10	96	105
6	NUM	SAGE	10	106	115
7	TEXT	SGENDER	10	116	125
8	TEXT	SELNAME	50	126	175
9	NUM	RANDOMNU	15	176	190
10	NUM	HH1	10	191	200
11	NUM	HH2	10	201	210
12	NUM	HH3	10	211	220
13	TEXT	HH4a001	255	221	475
14	TEXT	HH4a002	255	476	730

You can edit the Generated Variable List (the grid on the right of the screen) to change the widths of item on a one-by-one basis if desired. If you edit the list, you can tap **Edit and Apply Edits to Variable List** and the edits will be applied. If you wish to save the variable list to a file, tap **Save Variable List** and the utility prompts you for a storage location for the file. It saves the list as a comma-delimited file with six variables: sequence number, variable type, variable name, width, start column, and end column.

Sample of variable list file:

```
1,CASEID,CASEID,25,1,25
2,TEXT,USERID,20,26,45
3,TSTAMP,APPSTART,25,46,70
4,TSTAMP,SURVEYEXIT,25,71,95
5,NUM,SELECTEE,19,96,105
.
.
.
.
```

Finally, to create the flat or CSV file, tap **Create Flat or CSV File**. This action creates a file using the variable list derived earlier. It makes one record for each unique ID in the responses table for the selected survey, 0 or 1. The method for making the transpose is as follows:

- **Step 1:** Select the distinct IDs in the aggregated file.
- **Step 2:** Then for each ID found in Step 1, select the valid data and sort by QID, Date Time, prikey (an auto-number sequence in the responses table).
- **Step 3:** For each QID variable found in Step 2, place each variable in the designated record spaces as dictated by the variable list, and overwrite older data with the latest data if there are multiple instances of data for the same QID.
- **Step 4:** If a CSV file is desired, convert the fields in the flat record to CSV format for output.

The actual SQL select for this operation is as follows:

"Select * from responses where CaseID='" & ThisId & "' and valid = 1 order by QID, dttm, prikey"

where ThisId is a specific Case ID. After you tap **Create Flat or CSV File**, the system prompts you for a place to store the new file and then where to find the input SDF file that contains the data you are transposing.

The Help file for the transpose utility is repeated below.

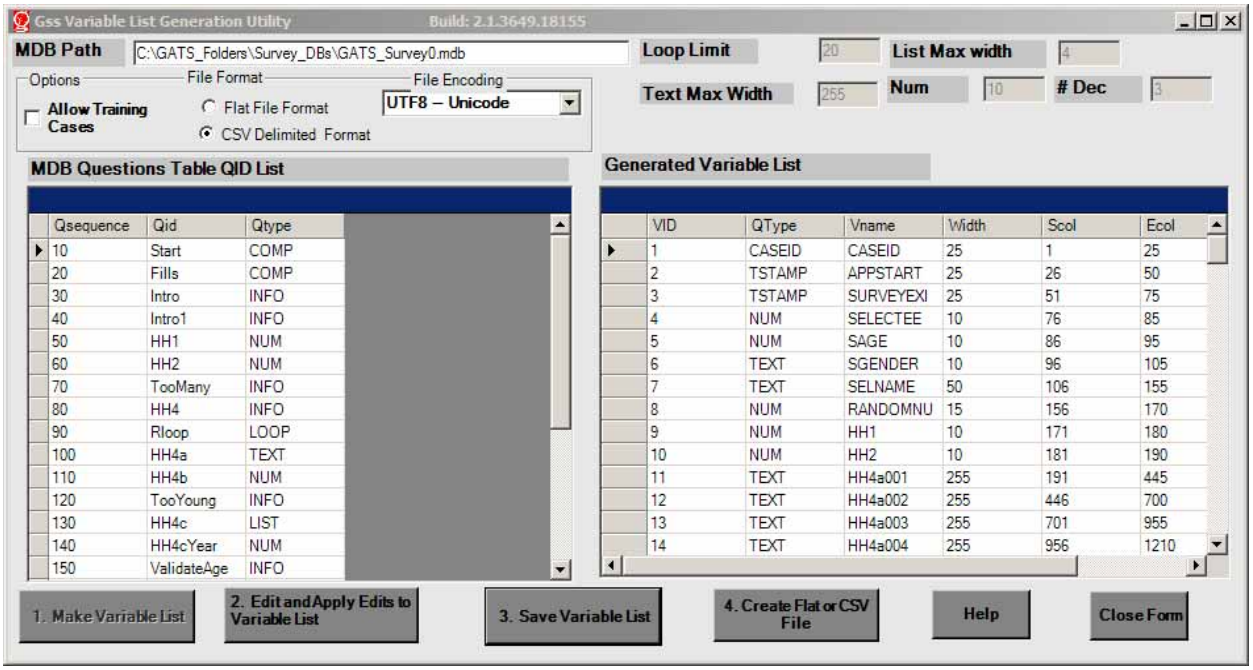
Flat or CSV File Generation

This utility is designed to reformat the data in the GSS Responses table (the answers to questionnaire questions), in individual SDF files or aggregated SDF files, into a form that is easier for analysis software to process. The GSS Responses table stores one row for every answer to a question and it contains backed-over data as well as current data. This format is the complete data record and audit trail but is hard to use in standard data analysis packages like SAS or SPSS. This utility allows users to convert the responses data into a format that is easier to work with in SAS or SPSS. It transposes the question by question row-oriented format of the responses table into a distinct flat record for each case. That is, it makes a single record, for each case, by taking the **most recent, valid response** to each question. If a question was not asked, the field for it is filled with blanks. The utility is driven by the MDB file for the GSS questionnaire specifications for either Survey0 or Survey1. That is, the utility uses the GSS MDB questionnaire specifications to create a list of all possible questionnaire variables and variable types. The screenshot in **Exhibit 5-8**, shows the startup screen. The Help button on this screen invokes this help file.

You can select the format for the output file (flat file or CSV delimited) and the data encoding (various forms of Unicode [UTF7, UTF8, UTF16 little or big Endian, UTF32] or ANSI). The flat file format is one single character record with fixed=width fields defined according to the variable list in the first step of the program. The CSV delimited format is comma-separated fields, one field for each row in the variable list. Text fields with embedded commas have leading and trailing “ (double quote) characters added to them. Text fields that have embedded “ (double quote) characters in them have the “ removed. The default format is CSV delimited and the default encoding is UTF8. Unicode (UTF8) and CSV delimited format will work in SPSS while either format will work in SAS. Unicode is required for character sets that require 16-bit characters (e.g., Russian and Chinese) and we have tested UTF8 format in SAS and SPSS.

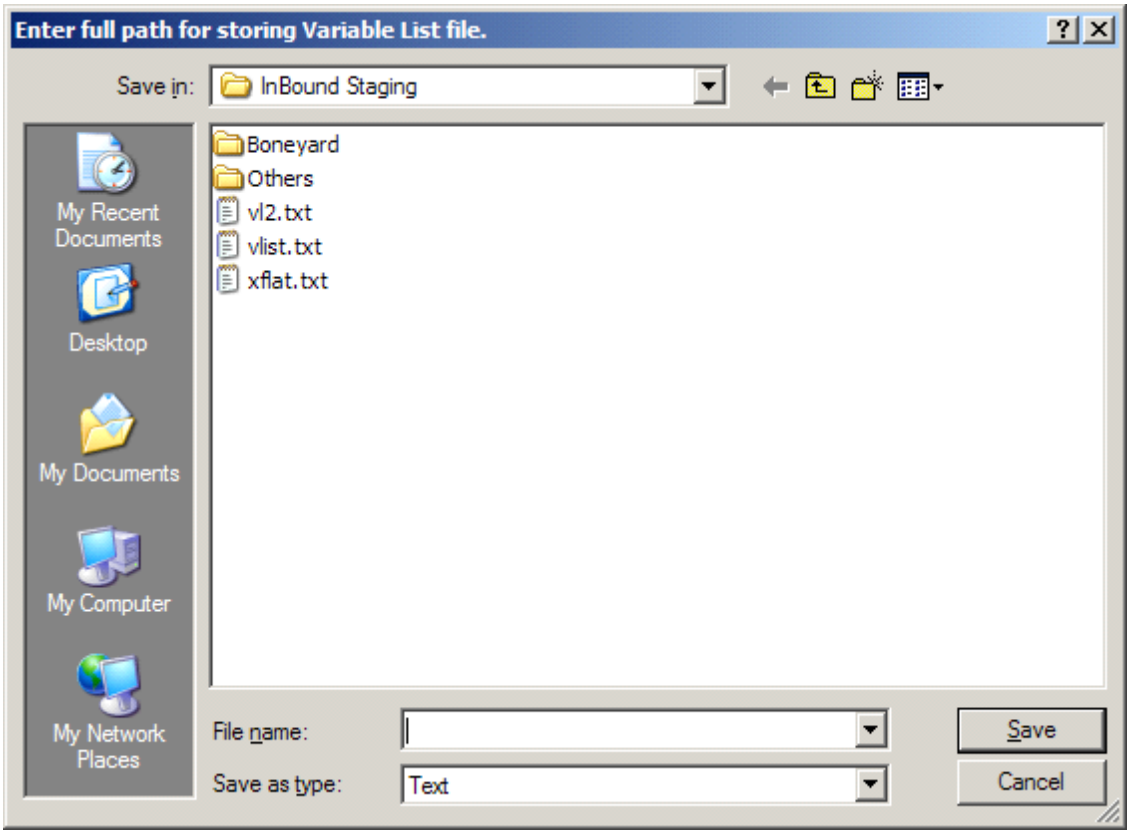
You can specify the limits and maximums for loops, text question field widths, list question field widths, and width and number of decimals for numeric questions. You can also allow training case data or not depending upon the state of the checkbox Allow Training Cases. If checked, training cases (IDs that start with XX) will be processed; otherwise, they will be ignored.

Exhibit 5-8. GSS Variable List Utility



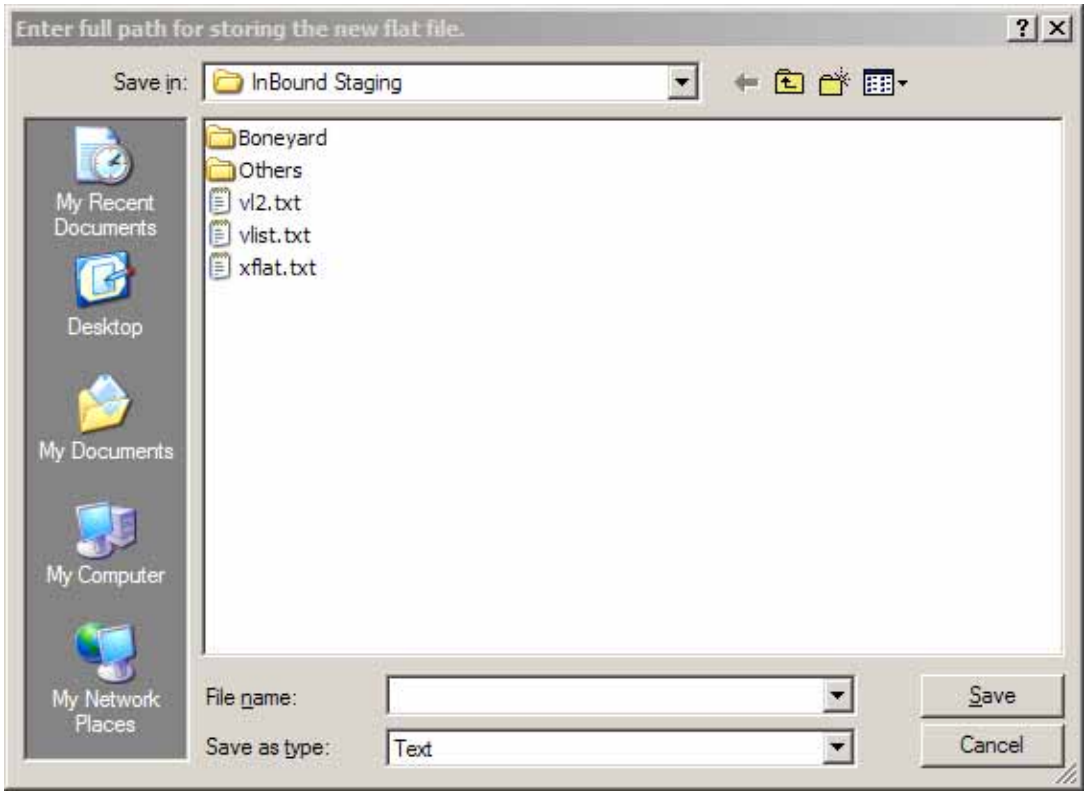
This software will generate two files for the user. It will generate a variable list from the input MDB file that is referenced in the MDB Path box. You can do this by tapping the **Make Variable List** button. This variable list can be edited to change widths of the columns. You can save the variable list to a file to use for reference or input to statistical software macros by tapping the **Save Variable List** button. After tapping this button you will be asked for a file location to store the variable list as shown in **Exhibit 5-9**.

Exhibit 5-9. Request for Path for Storing Variable List (.txt) File



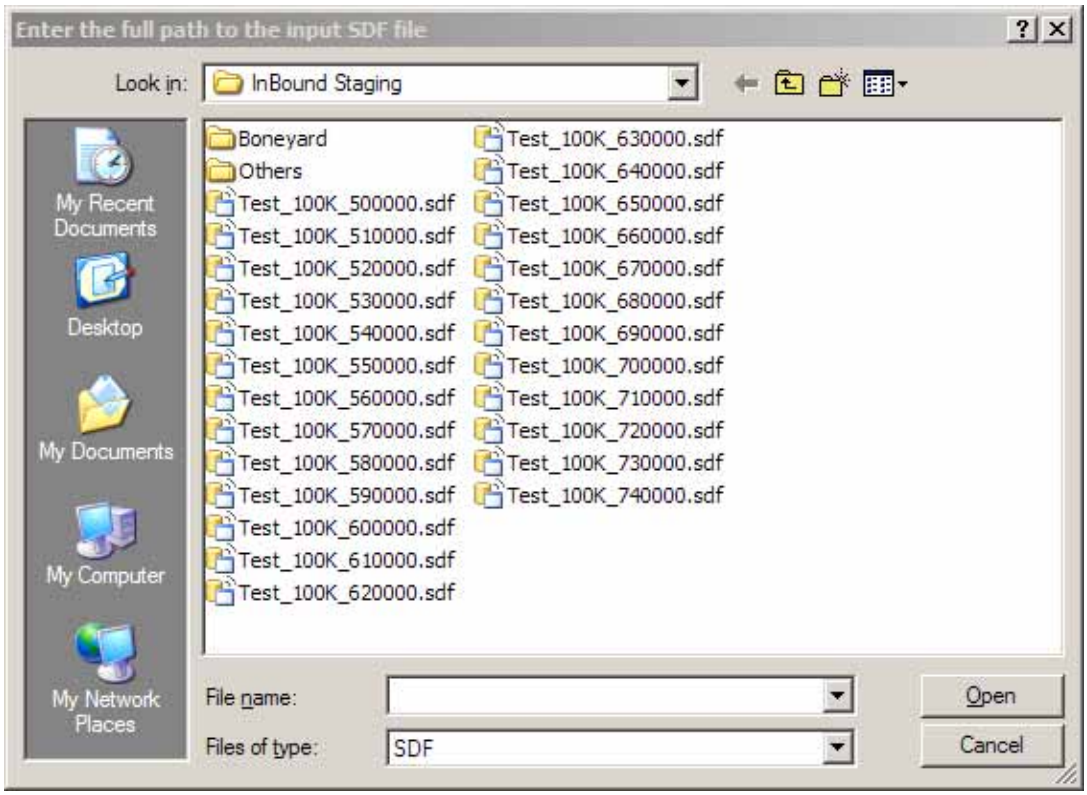
Once edits are complete, you can ask the tool to create a transposed file from a selected SDF file using the variable list that you have just generated as a guide or driver for the file creation. You do this by tapping the **Create Flat or CSV File** button. The screen shown in **Exhibit 5-10** demonstrates this path.

Exhibit 5-10. Request for Path for Storing Output Data File (Flat File or CSV Delimited File)



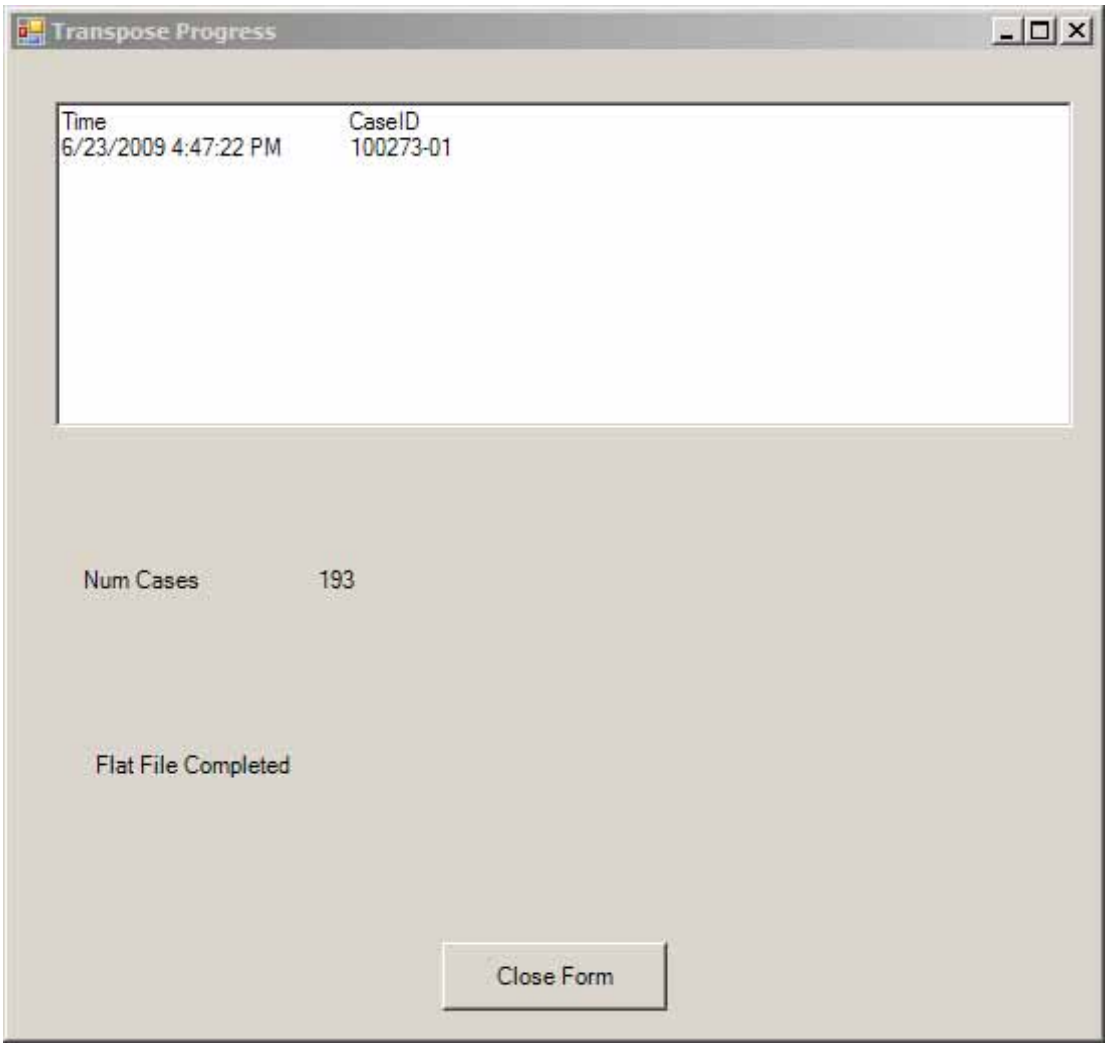
And finally, the program prompts you for the location of an SDF file to use as the source or input data to generate the transposed. **Exhibit 5-11** shows this file path request.

Exhibit 5-11. Request for Path to Input SDF File with Raw Data



While the transpose is in progress, it will display a case count (updated every 300 cases). Once the output file has been created, it will display this final summary screen. The screen in **Exhibit 5-12** shows that 193 cases have been created and the last CASEID is 100273-01. The user should expect a file of 10,000 cases to take 10–15 minutes to complete on a typical laptop.

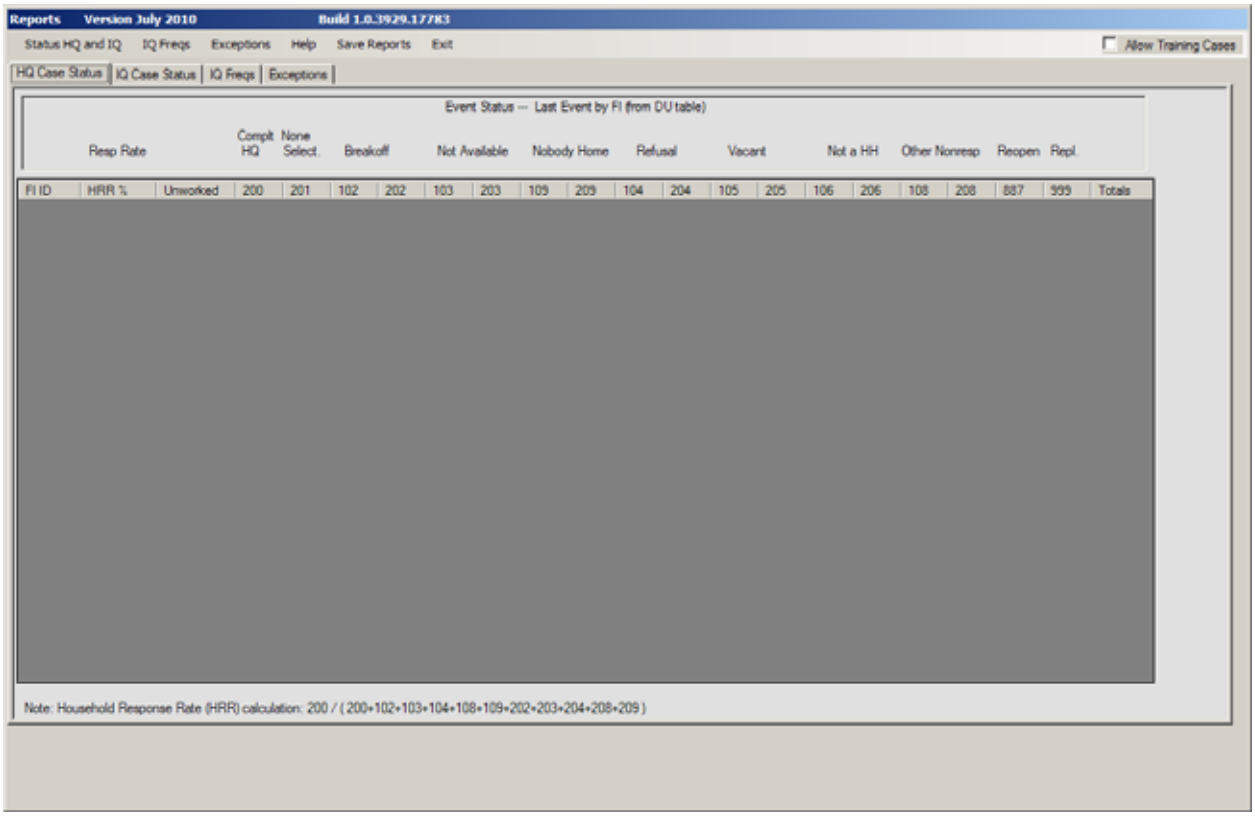
Exhibit 5-12. Transpose Summary Screen



Reporting Utility

The Data Aggregation menu contains a reporting tool that allows users to monitor the content of an aggregated data file for status monitoring and quality control purposes.

Exhibit 5-12a. Reporting Utility



The Reporting menu option brings up a standalone utility. This utility allows the GATS manager to create three different status or data reports to view on screen or export to a CSV or text file. The user selects the desired report by tapping on one of the Menu items for reporting—Status HQ and IQ, IQ Freqs, or Exceptions. The user will be asked to identify or browse the required input files. The user should point the utility to the Master Aggregated SDF file that he/she has made by aggregating the individual SDF files from the Field Interviewer handhelds. In addition to the Master Aggregated SDF file, the PC that executes these programs will need current versions of the MDB files that define the surveys and will need to identify the Primary Sampling Unit or Primary Sampling Unit variable in the case DU table. The MDB files will be used to produce labels and formats for the reports while the PSU variable can be used as a "by" variable for frequencies.

The three reports that are available are:

- **Case Status Report**—a Survey Event Status Report based on the latest event or Record of Call code (from the DU table), that details that status of each Field Interviewer's work load (see **Exhibit 5-13** for an example).
- **IQ Frequencies Report**—the second report option allows GATS data managers to request descriptive frequency distributions of the data from completed *Individual Questionnaire*s. This report

will allow managers to use a report filter that controls the level of the report from drop down lists of PSUs and or Field Interviewers. Drop-down lists will allow data managers to select up to 20 of the *Individual Questionnaire* variables for a given report, and to select one of the existing survey languages to use for the labels of question text and response stub text for the report (see **Exhibit 5-14** for screenshot of the filter screen and **Exhibit 5-15** for an example of the frequency report).

- **Exceptions Report**—the third report lists Field Interviewers who have exceeded a specified number of call records for a case but have not coded the case as final (see **Exhibit 5-16** for an example). This report is based on the result codes in the aggregated DUEVT table. If a Field Interviewer has a case that has >= four record of calls records but is still not in a final status, then that case will be listed in the report. This report will detail separately cases for the *Household Questionnaire* (HH) events and the *Individual Questionnaire* (IQ) events. If cases have HH events but no IQ events, the IQ columns will be blank.

Exhibit 5-13. Survey Event Status Report HQ Case Status

HQ Case Status | IQ Case Status | IQ Freqs | Exceptions

Event Status --- Last Event by FI (from DU table)

Resp Rate	Compl HQ	None Select.	Breakoff	Not Available	Nobody Home	Refusal	Vacant	Not a HH	Other Nonresp	Reopen	Repl.

FI ID	HRR %	Unworked	200	201	102	202	103	203	109	209	104	204	105	205	106	206	108	208	887	999	Totals
987001	100.00		40																		40
987002	100.00		20																		20
987003	100.00		20																		20
987004	100.00		20																		20
987005	100.00		20																		20
987006	100.00		20																		20
987007	100.00		19	1																	20
987008	100.00		20																		20
987009	100.00		20																		20
987010	100.00		39	1																	40
987011	100.00		20																		20
987012	100.00		20																		20
987013	100.00		19	1																	20
987014	100.00		20																		20
987015	95.00		38									1						1			40
987016	95.00		19															1			20
987017	95.00		19															1			20
987018	100.00		20																		20
987019	100.00		20																		20

Note: Household Response Rate (HRR) calculation: 200 / (200+102+103+104+108+109+202+203+204+208+209)

Exhibit 5-14. Filter Screen

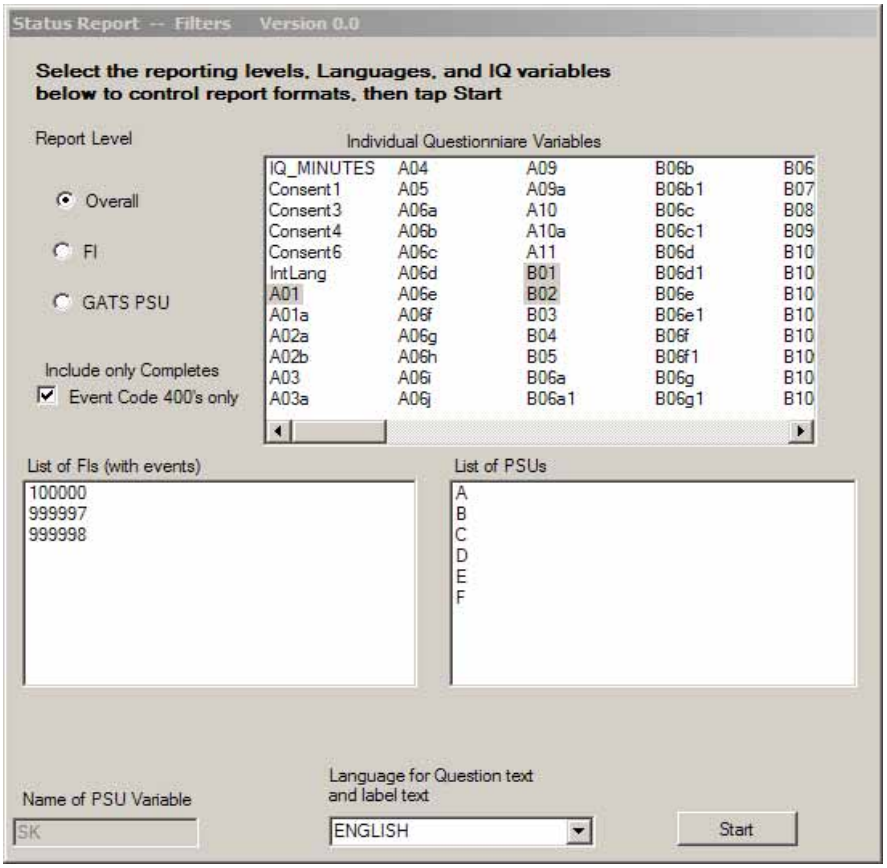


Exhibit 5-15. Questionnaire Frequency Distributions

Individual Questionnaire Frequency Distributions (Completed Interviews Only – code 400)

```
OVERALL        Question A01   Question Type: LIST                    7/27/2010 09:48:15

RECORD GENDER FROM OBSERVATION.  ASK IF NECESSARY.

Code        Code Text                              Freq.          %              % (no Missing)
1           MALE                                   6224          54.47          54.47
2           FEMALE                                 5203          45.53          45.53
            Missing                                                .00
Total   ........................................11427

_____________________________________________________________________

OVERALL        Question B01   Question Type: LIST                    7/27/2010 09:48:15

Do you *currently* smoke tobacco on a daily basis, less than daily, or not at all?

Code        Code Text                              Freq.          %              % (no Missing)
1           DAILY                                  4272          37.39          37.39
2           LESS THAN DAILY                         537           4.70           4.70
3           NOT AT ALL                             6615          57.89          57.89
9           REFUSED                                   1            .01            .01
7           DON'T KNOW                                2            .02            .02
```

Exhibit 5-16. FI Exceptions Cases with No Final Codes and >= Four Result Codes

Exception List

Cases with >=4 Result Ciodes and no final code

Most recent 5 result codes per Case; HH= household events; IQ= Individual events

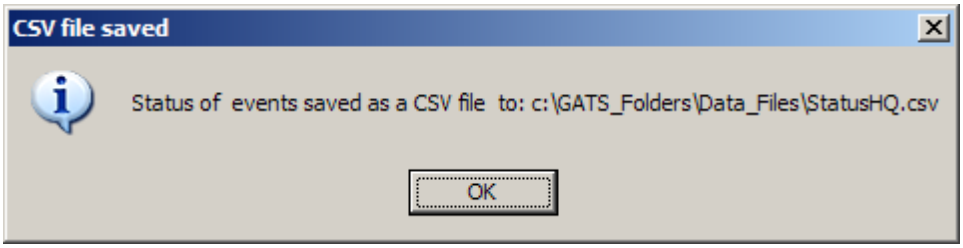
FI ID	Case ID	HH Event Count	HH 1	HH 2	HH 3	HH 4	HH 5	IQ Event Count	IQ 1	IQ 2	IQ 3	IQ 4	IQ 5
987113	123970-00	4		109 7/11/2009	109 7/14/2009	109 7/18/2009	109 7/22/2009	0					

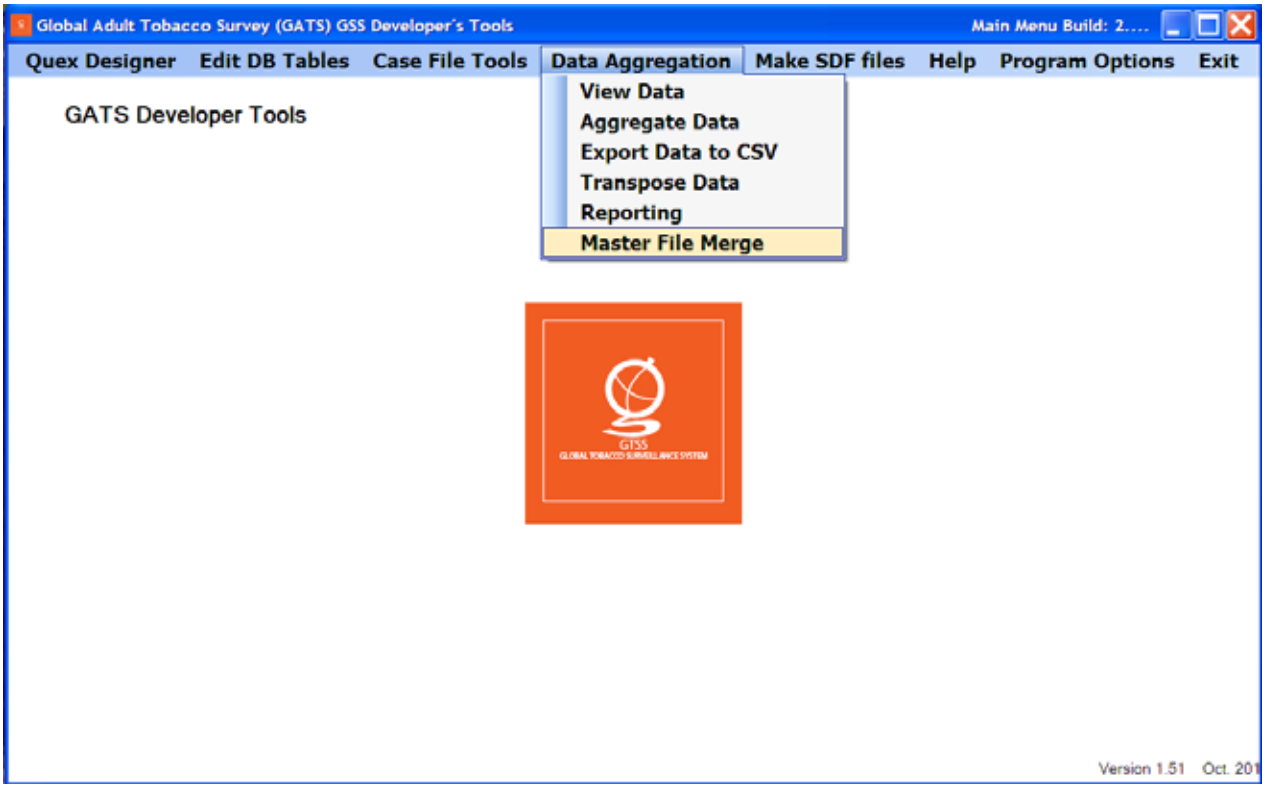
The user can request Help for this utility from the Help menu item. Also the user can save the reports he generates by using the Save Reports menu item. If the user clicks on the Save Reports menu item a list of reports that can be saved is shown. If a report name is grayed out it means it has not been run yet so is not available for saving. The saved files will be stored in the C:\Gats_Folders\Data_files folder as CSV or text files that can be printed or imported into other tools like Excel.

The data manager(s) should use these tools to review and evaluate the survey status and data as survey operations proceed. The manager(s) should be reviewing the aggregated SDF data files at the level of aggregation for which they are responsible (region or state or nation) to ensure the survey is on schedule and the data collected meet quality standards.

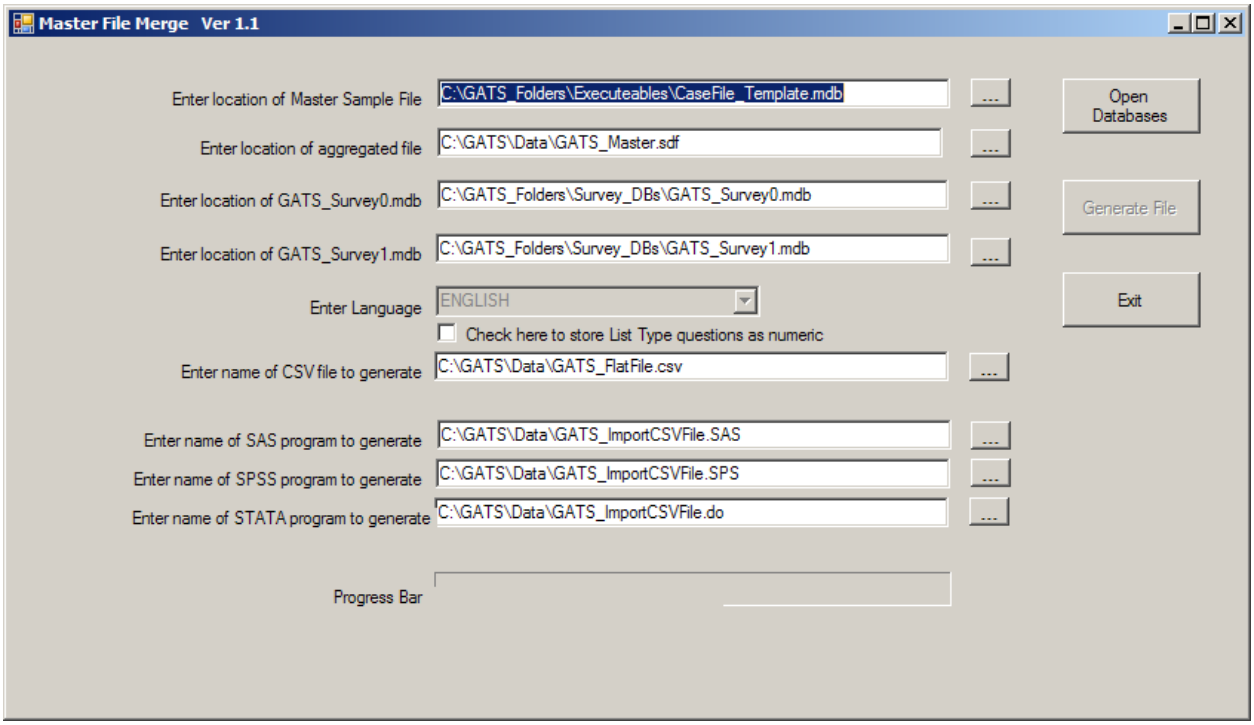
Master File Merge

The Master File Merge option combines study data into a single, comma delimited file and creates SAS, SPSS and Stata input programs capable of reading the newly merged file. To use the Master File Merge, select Master File Merge from the Data Aggregation menu item (as shown in the following screenshot).





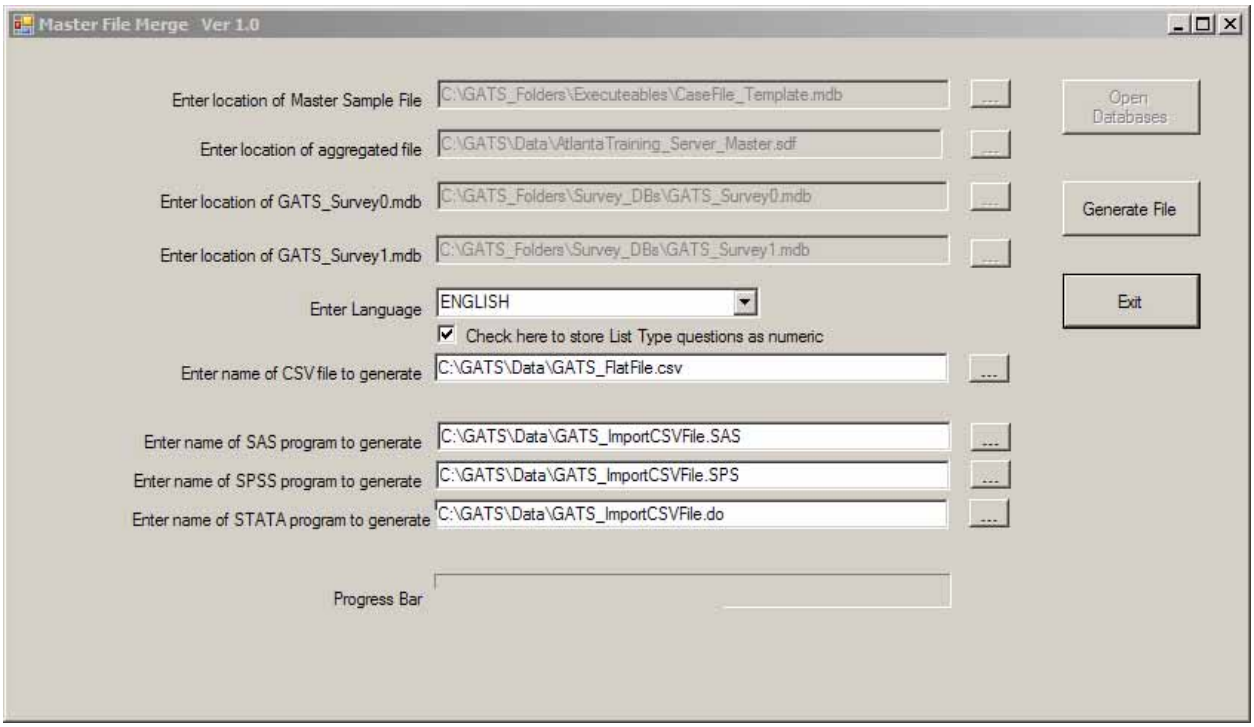
The system has default values for the location of the required files. The files used in the Master File Merge include the Master Sample File, the aggregated SDF file, and the two questionnaire databases (GATS_Survey0.mdb and GATS_Survey1.mdb). The Master Sample File is the MS Access database used to create the case file for the handheld. Additional fields may be added to this database, but the field names, types, and sizes should not be updated. The field description of fields used by your site should be updated so that the descriptive labels produced by the statistical programs are correct (see following screenshot).



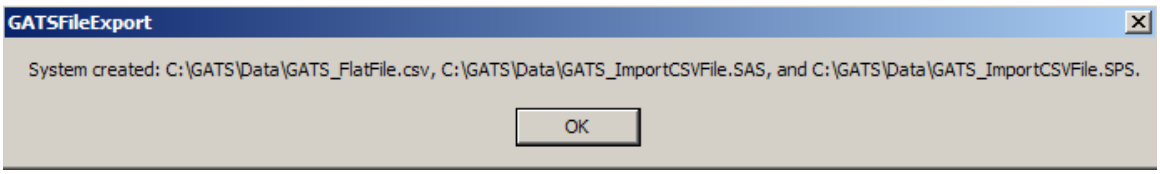
Start by updating the file locations of the master case file, the aggregated file, and the survey questionnaire databases. To change, update the file location or name in the text box or select the "…" button and the system will allow you to select an existing file. Once the files locations and names are correct, select the **Open Databases** button. This will check the file locations and ensure that the system can open the databases. It also gathers information about languages that are available.

After you have pressed the **Open Databases** button, the databases are open and the option to change or open other databases is not available. You can now select the language for SAS, SPSS and Stata formatted data to display. At any time, you can select the folder and name of the CSV file, the SAS, Stata and the SPSS input programs that are created.

There is a check box that allows the user to store list type questions as numeric. If this option is checked all list type questions will be stored as numeric. Note that if character has been stored in these fields the SAS, SPSS, and Stata programs will need to be modified to run properly.



Click the **Generate File** button to create the CSV, SAS, SPSS and Stata files. The CSV will be a comma delimited file in Unicode (UTF-8) format. This program may take several minutes to run. The hourglass mouse pointer will display while the program is running. Once it is completed, the following message is displayed.



To view the CSV file, a Unicode-enabled text editor such as WordPad is required. The SAS and SPSS files are also in Unicode to ensure that labels and formats are displayed correctly. SAS users should run the program using the SAS Unicode Server. Stata does not support Unicode but will read the comma delimited file. Text in labels and data that are non-western character sets may not display correctly. The statistical programs may need to be updated prior to running them to ensure that country specific options are set.

6. Files and Structures

The CMS and the GSS engine require a Windows CE compatible database for operation. RTI has chosen to use Microsoft SQL Server 2005 Compact Edition (SQL CE) files on the handheld and desktop. The same format file can be used on the handheld and desktop. A strong feature of this type of database is that it can also be used without conversion on a desktop system running Windows XP or later. Users can either build and maintain files in a standard MS SQL Server database and import them to the Pocket PC or, as we have chosen, they can build and maintain the files in MS Access on a desktop PC and use a conversion utility to move them from MS Access MDB files to SQL CE SDF files. In addition, Microsoft provides SQL CE tools for directly managing SQL CE files on a PC (Microsoft SQL Server Management Studio Express).

The GSS requires a minimum of two SQL Server CE databases for a given data collection activity, one for the CMS system and one for each GSS questionnaire. These files must be located in the iPAQ in a root folder named ProjectName Programs, where the project name is a character string that the user decides to use as a project name. If the project name is ABCD, then the folder name for all the project files would be ABCD Programs. The database files start with a project name, then the underscore character, and then CMSDB for the CMS and SurveyX, where X is 0,1,2, …n, and n is the number of survey forms −1. The default naming convention for these two database files for a survey with project name ABCD and with only one survey form would be as follows:

- CMS database: ABCD_CMSDB.sdf
- GSS database: ABCD_Survey0.sdf

If there are multiple survey forms, then additional ProjectName_SurveyX.sdf files are needed. In GATS, the files names are:

- GATS_CMSDB.sdf
- GATS_Survey0.sdf (the *Household Questionnaire*)
- GATS_Survey1.sdf (The *Individual Questionnaire*)

The SQL CE database used by the CMS contains multiple SQL accessible data tables that provide the following:

- startup parameters and training data for the CMS (case information, startup data)
  Tables: INI, DU, TrainingCases
- storage tables for control information for the CMS
  Tables: Delrecs, EvtDef, InterviewInfo, Messages
- storage tables for CMS data provided by a Field Interviewer or CMS system
  Tables: Addresslog, DuEvt, Tasks, Verinfo, FSTable, PTE
- storage tables for GSS data provided by the Field Interviewer (question answers, notes)
  Tables: Responses, Notes

The following sections detail the design and definitions of the tables used in the SQL CE databases. The data types and sizes are based on the Access format before conversion to SQL CE tables. A final section details how to use a standalone iPAQ utility maintain SDF files and to repair corrupted SDF files.

Exhibit 6-1 details the contents of the CMSDB.sdf database and describes the purpose of each internal table. Detailed information about the columns within each table and their key structure is provided in **Exhibits 6-2** through **6-15**. **Chapters 4** and **7** provide additional detail about the use of these tables and the data fields. **Appendix B** provides additional information and some examples of the main data files in the CMSDB of interest to the analyst (Responses, DUEvt, DU).

Exhibit 6-1. Purpose of Internal Tables of CMSDB.sdf

Table Name	Purpose	Required for GATS	Read/ Write
Addresslog	Stores address changes and edits made by Field Interviewers.	Yes	R/W
Delrecs	Stores requests from the transmission system for cases to be removed from the iPAQ to be transferred by another Field Interviewer (assumes network data transfer mode).	Optional	R/W
DU	The basic case table where each row represents a case to be worked and tracked by a Field Interviewer. It is loaded with cases at the start of the survey, and cases can be added or removed during survey operations. Contains sample and locator information.	Yes	R/W
DUEvt	Stores event data recorded by Field Interviewers while they work and track their cases.	Yes	R/W
EvtDef	Stores a read-only list of possible events for each form.	Yes	R
FSTable	Stores a read-only list of Field Supervisors and their Field Interviewers.	Optional	R
INI	Stores lists of start-up parameters.	Yes	R/W
InterviewInfo	A temporary data record that is passed between the CMS and GSS to transfer data from the CMS to GSS to start an interview. Data are live only during the transfer.	Yes	Temp
Messages	Stores label, error, and information messages in multiple languages.	Yes	R
Notes	Stores note information, by case, as input by Field Interviewers. Serves as a case diary for informal notes about the case.	Yes	R/W
PTE	Not used by GATS.	No	R/W
Responses	Stores the response data collected during questionnaire administration in GSS, one row for each question that collects data.	Yes	R/W
Tasks	Not used in this version of GSS.	No	R/W
TrainingCases	Test cases to be used for training and Field Interviewer practice. Same format and structure as the DU table. ALL training IDs start with XX to distinguish them from live case data in the DU table.	Yes	R/W
Verinfo	Not used in this version of GATS.	No	R/W

Exhibit 6-2. CMS Database Table: AddressLOG

Name	Type	Size
CASEID	Text	25
NumCaseID	Text	14
CREATEDATE	Date/Time	8
STREETNO_O	Text	10
STREET_O	Text	55
APT_NUM_O	Text	10
CITY_O	Text	25
STATE_O	Text	25
ZIPCODE_O	Text	10
BOX_NUM_O	Text	25
RURAL_ROUTE_O	Text	25
STREETNO_N	Text	10
STREET_N	Text	55
APT_NUM_N	Text	10
CITY_N	Text	25
STATE_N	Text	25
ZIPCODE_N	Text	10
BOX_NUM_N	Text	25
RURAL_ROUTE_N	Text	25
AUTH_CODE	Text	10
POSTED	Text	1
POSTDTTM	Date/Time	8
TZ	Integer	2
TZ1	Integer	2

Table Indexes

Name	Number of Fields
AUTH_CODE	1
Fields:	
AUTH_CODE	Ascending
PK_AddressLog	2
Fields:	
CASEID	Ascending
CREATEDATE	Ascending
Posted	1
Fields:	
POSTED	Ascending

Exhibit 6-3. CMS Database Table: Delrecs

Name	Type	Size
CASEID	Text	25
NumCaseID	Text	14
CreateDtTm	Date/Time	8
SendTo	Text	10
Sent	Yes/No	1
SentTS	Text	50
Received	Yes/No	1
ReceivedDtTm	Date/Time	8

Table Indexes

Name	Number of Fields
NumCaseID	1
Fields:	
NumCaseID	Ascending
PK_Delrecs	3
Fields:	
CASEID	Ascending
CreateDtTm	Ascending
SendTo	Ascending
SendToAndSent	2
Fields:	
SendTo	Ascending
Sent	Ascending

Exhibit 6-4. CMS Database Table: DU

Name	Type	Size
CASEID	Text	25
NumCaseID	Text	14
FormNum	Text	2
ProjectName	Text	20
SCRNUM	Integer	2
CREATEDATE	Date/Time	8
STREETNO	Text	10
STREET	Text	55
APT_NUM	Text	10
BOX_NUM	Text	10
RURAL_ROUTE	Text	10
TRACT_BG	Text	25
BLOCK	Text	25
HC	Text	25
SK	Text	25
TYPE	Text	25
CITY	Text	25
STATE	Text	25
ZIPCODE	Text	10
COUNTY	Text	25
EVTCODE	Text	50
PERIOD	Integer	2
YEAR	Text	50
WAVE	Integer	2
ROCBASE	Integer	2
TRAININGFLAG	Text	1
FSID	Integer	2
ALTERNATIVE	Text	1
ACTIVE	Text	1
ALTER_CASEID	Text	25
PostDTTM	Date/Time	8
POSTED	Text	1
Name	Text	50
Phone	Text	50
OtherID	Text	50

Table Indexes

Name		Number of Fields
ALTER_CASEID		1
	Fields:	
	ALTER_CASEID	Ascending
caseid		1
	Fields:	
	CASEID	Ascending
CITY		1
	Fields:	
	CITY	Ascending

Exhibit 6-5. CMS Database Table: DUEvt

Name	Type	Size
CASEID	Text	25
NumCaseID	Text	14
CREATEDATE	Date/Time	8
CONT_NO	Integer	2
SCRNUM	Integer	2
EVENTDATE	Date/Time	8
RESULT	Text	50
FIID	Text	8
COMMENTS	Memo	—
HISPBOX	Integer	2
MODIFYDTTM	Date/Time	8
AUTO_EVENT	Text	1
NULLIFIED	Text	1
AUTH_CODE	Text	5
RECVDTTM	Date/Time	8
POSTDTTM	Date/Time	8
POSTED	Text	1
TZ	Integer	2
TZ1	Integer	2

Table Indexes

Name		Number of Fields
AUTH_CODE		1
	Fields:	
	AUTH_CODE	Ascending
CaseID		1
	Fields:	
	CASEID	Ascending
NumCaseID		1
	Fields:	
	NumCaseID	Ascending
PK_DUEvt		4
	Fields:	
	NumCaseID	Ascending
	CREATEDATE	Ascending
	CONT_NO	Ascending
	SCRNUM	Ascending
Posted		1
	Fields:	
	POSTED	Ascending

Exhibit 6-6. CMS Database Table: EvtDef

Name	Type	Size
EvtCode	Integer	2
EvtDesc	Text	255
Status	Double	8
DTRequired	Text	1
AvailableToFI	Text	1
AllowUpdate	Text	1
Task	Text	2

Table Indexes

Name	Number of Fields
PriKey	2
Fields:	
EvtCode	Ascending
Task	Ascending
PrimaryKey	2
Fields:	
EvtCode	Ascending
Task	Ascending

Exhibit 6-7. CMS Database Table: FSTable

Name	Type	Size
FSID	Text	10
FSName	Text	50

Table Indexes

Name	Number of Fields
PK_FSTABLE	1
Fields:	
FSID	Ascending

Exhibit 6-8. CMS Database Table: INI

Name	Type	Size
VariableName	Text	20
VariableValue	Text	255
Comment	Text	255

Table Indexes

Name	Number of Fields
PK_Ini	1
Fields:	
VariableName	Ascending

Exhibit 6-9. CMS Database Table: InterviewInfo

Name	Type	Size
CASEID	Text	25
NumCaseID	Text	14
SCRNUM	Integer	2
STREETNO	Text	10
STREET	Text	55
APT_NUM	Text	10
CITY	Text	25
STATE	Text	25
ZIPCODE	Text	10
CONT_NO	Integer	2
FIID	Text	10
FIName	Text	50
CURRENT_EVT_CODE	Text	50
RR	Text	10
Box	Text	20
Modify_Flag	Yes/No	1
AuthFlagRequired	Yes/No	1
County	Text	25
EventAdded	Integer	2
Parm2	Text	50
Parm3	Text	50
Parm4	Text	50
Parm5	Text	50
Parm6	Text	50
FormNum	Text	2
ProjectName	Text	20

Table Indexes

Name	Number of Fields
FormNum	1
Fields:	
FormNum	Ascending
NumCaseID	1
Fields:	
NumCaseID	Ascending
PK_InterviewInfo	1
Fields:	
CASEID	Ascending

Exhibit 6-10. CMS Database Table: Messages

Name	Type	Size
PK	Long Integer	4
MsgId	Text	25
LanguageID	Text	20
Message	Text	255

Table Indexes

Name		Number of Fields
LanguageID		1
	Fields:	
	LanguageID	Ascending
MsgId		1
	Fields:	
	MsgId	Ascending
PrimaryKey		1
	Fields:	
	PK	Ascending

Exhibit 6-11. CMS Database Table: Notes

Name	Type	Size
CaseID	Text	25
NumCaseID	Text	14
QID	Text	20
Note	Memo	—
DtTm	Date/Time	8
Version	Text	50
UserId	Text	20
PriKey	Long Integer	4
Posted	Text	1
PostedDtTm	Date/Time	8
TZ	Integer	2
TZ1	Integer	2

Table Indexes

Name		Number of Fields
CaseID		1
	Fields:	
	CaseID	Ascending
NumCaseID		1
	Fields:	
	NumCaseID	Ascending
PK_Notes		1
	Fields:	
	PriKey	Ascending
Posted		1
	Fields:	
	Posted	Ascending

Exhibit 6-12. CMS Database Table: Responses

Name	Type	Size
CaseID	Text	25
NumCaseID	Text	14
QID	Text	20
Answer	Text	255
Instance	Long Integer	4
Lang	Integer	2
DtTm	Date/Time	8
Version	Text	50
UserId	Text	20
PriKey	Long Integer	4
Valid	Yes/No	1
OnHold	Text	1
Posted	Text	1
PostDtTm	Date/Time	8
TZ	Integer	2
TZ1	Integer	2

Table Indexes

Name	Number of Fields
CaseID	1
Fields:	
CaseID	Ascending
CaseIDAndPosted	2
Fields:	
CaseID	Ascending
Posted	Ascending
DateTime	1
Fields:	
DtTm	Ascending
NumCaseID	1
Fields:	
NumCaseID	Ascending
PK_Responses	1
Fields:	
PriKey	Ascending
Posted	1
Fields:	
Posted	Ascending

Exhibit 6-13. CMS Database Table: Tasks

Name	Type	Size
TaskNum	Text	25
TaskName	Text	50

Table Indexes

Name	Number of Fields
PK_Tasks	1
Fields:	
TaskNum	Ascending

Exhibit 6-14. CMS Database Table: TrainingCases

Name	Type	Size
CASEID	Text	25
NumCaseID	Text	14
FormNum	Text	2
ProjectName	Text	20
SCRNUM	Integer	2
CREATEDATE	Date/Time	8
STREETNO	Text	10
STREET	Text	55
APT_NUM	Text	10
BOX_NUM	Text	25
RURAL_ROUTE	Text	25
TRACT_BG	Text	25
BLOCK	Text	25
HC	Text	25
SK	Text	25
TYPE	Text	25
CITY	Text	25
STATE	Text	25
ZIPCODE	Text	10
COUNTY	Text	25
EVTCODE	Text	50
PERIOD	Integer	2
YEAR	Text	50
WAVE	Integer	2
ROCBASE	Integer	2

Exhibit 6-14. CMS Database Table: TrainingCases (continued)

Name	Type	Size
TRAININGFLAG	Text	1
FSID	Integer	2
ALTERNATIVE	Text	1
ACTIVE	Text	1
ALTER_CASEID	Text	25
POSTDTTM	Date/Time	8
POSTED	Text	1

Table Indexes

Name		Number of Fields
OtherID	1	
	Fields:	
	OtherID	Ascending
PK_PrimaryKey		1
	Fields:	
	NumCaseID	Ascending

Exhibit 6-15. CMS Database Table: VERINFO

Name	Type	Size
CASEID	Text	25
NumCaseID	Text	14
CREATEDATE	Date/Time	8
SCRNUM	Integer	2
CONT_NO	Text	50
NAME	Text	30
PHONE	Text	15
TIME_PERIOD	Text	200
FIID	Text	50
POSTED	Text	1
POSTDTTM	Date/Time	8
TZ	Integer	2
TZ1	Integer	2

Table Indexes

Name		Number of Fields
NumCaseID		1
	Fields:	
	NumCaseID	Ascending
PK_Verinfo		3
	Fields:	
	CASEID	Ascending
	CREATEDATE	Ascending
	FIID	Ascending
Posted		1
	Fields:	
	POSTED	Ascending

The next section discusses the eight tables that are included in the survey form SDF database. The purpose of these tables is shown in **Exhibit 6-16**. A detailed listing of the columns and formats for each of the questionnaire databases is presented next. **Exhibits 6-17** through **6-20** show the list of the tables in the User Maintainable database, and **Exhibits 6-21** through **6-24** show the list of tables in the Static database. **Chapters 4** and **7** also discuss the contents and use of these tables.

Exhibit 6-16. Purpose of Questionnaire Database Tables

Table Name	Purpose	Required for GATS	Read/ Write
Answers	Stores answer sets that provide the answer choices for “list” or “all that apply” questions.	Yes	R
INI	Stores lists of startup parameters.	Yes	R/W
Questions	Contains the main metadata store that provides the data for the GSS questionnaire engine. Each question is a row in this table and provides the logic, validations, etc. for each question. The text for a question is in a separate table, Texts, which has a row for each QID and language.	Yes	R
QxQ	A read-only table of Help information that is provided for all or some of the questions. If QxQ help is available for a given screen, the QxQ menu item is enabled and, when tapped, will show the text provided in this table for a given question.	Optional	R
FormatType	Stores a read-only list of data types used by the special instructions fields of the Questions table.	Yes	R
Opers	Stores read-only list of operators for logical comparisons. Used only in older versions of the GSS.	No	R
Types	Stores a read-only list of the types of screens allowed by the GSS.	Yes	R
Texts	Holds the text for questions, one row for a QID and Language. There is a QID row for each language desired.	Yes	R

Exhibit 6-17. User Maintainable Database Tables for a Survey Form: Answers

Name	Type	Size
UID	Long Integer	4
AID	Text	20
ASequence	Long Integer	4
ACode	Text	20
AText	Text	255
AWav	Text	50

Table Indexes

Name		Number of Fields
ACode		1
	Fields:	
	ACode	Ascending
AID	1	
	Fields:	
	AID	Ascending
PrimaryKey		1
	Fields:	
	UID	Ascending

Exhibit 6-18. User Maintainable Database Tables for a Survey Form: INI

Name	Type	Size
VariableName	Text	20
VariableValue	Text	50
Comment	Text	255

Table Indexes

Name		Number of Fields
PK_Ini		1
	Fields:	
	VariableName	Ascending

Exhibit 6-19. User Maintainable Database Tables for a Survey Form: Questions

Name	Type	Size
QSequence	Single	4
Qid	Text	20
QText	Memo	—
Qwav	Text	255
QNext	Text	20
Qlogic	Memo	—
QType	Text	8
Format	Text	255
Qanswer	Text	20
RangeLo	Single	4
RangeHi	Single	4
LoopStart	Text	20
LoopEnd	Text	20
LoopGoTo	Text	20
RosterNumber	Integer	2
RosterCol	Integer	2

Table Indexes

Name	Number of Fields
PK_Questions	1
Fields:	
Qid	Ascending
Qanswer1	
Fields:	
Qanswer	Ascending

Exhibit 6-20. User Maintainable Database Tables for a Survey Form

QxQ

Properties:

DateCreated:	12/21/2007 2:32:18 PM	LastUpdated:	12/21/2007 2:232:28PM
RecordCount:	29	Updatable:	True

Columns

Name	Type	Size
QID	Text	20
QxQText	Memo	—
QxQWav	Text	255

Table Indexes

Name	Number of Fields
PK_QxQ 1	
Fields:	
QID	Ascending

Texts

Properties:

DateCreated:	12/21/2007 2:32:18 PM	LastUpdated:	12/21/2007 2:232:28PM
RecordCount:	29	Updatable:	True

Columns

Name	Type	Size
QID	Text	20
QText	Memo	—
QWav	Text	255
Language	Text	25
Pkey	Long Integer	4

Table Indexes

Name	Number of Fields
PKey 1	
Fields:	
Pkey	Ascending

Exhibit 6-21. Static Database Tables for a Survey Form, Not to Be Modified by the User: FormatType

Name	Type	Size
FormatType	Text	10

Exhibit 6-22. Static Database Tables for a Survey Form, Not to Be Modified by the User: Opers

Name	Type	Size
Operator	Text	2

Exhibit 6-23. Static Database Tables for a Survey Form, Not to Be Modified by the User: Types

Name	Type	Size
FormatType	Text	4

Table Indexes

Name	Number of Fields
PK_Types	1
Fields:	
Types	Ascending

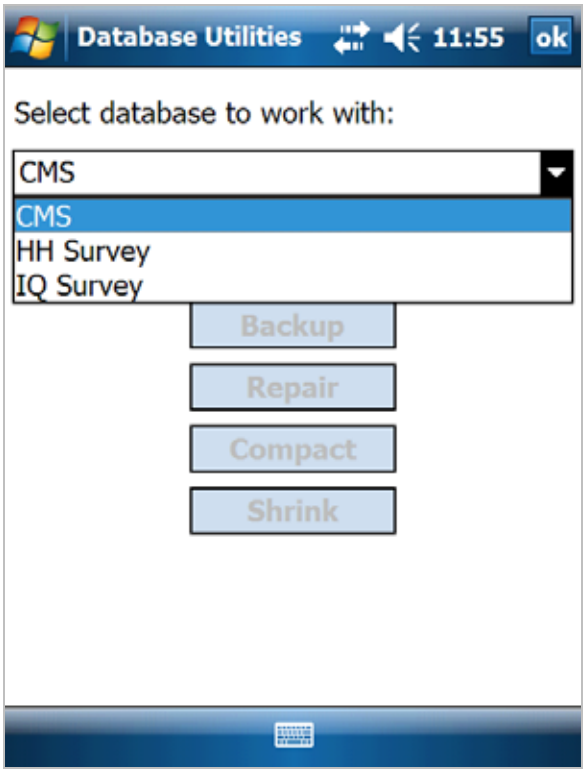
Exhibit 6-24. Optional Database Tables for a Survey Form: AnswerKeys

Name	Type	Size
CaseID	Text	25
NumCaseID	Text	50
QID	Text	20
Answer	Text	255
Instance	Long Integer	4
Lang	Long Integer	4
DtTm	Date/Time	8
UserID	Text	20
Version	Text	50
PriKey	Long Integer	4
Valid	Yes/No	1
OnHold	Text	1
Posted	Text	1
PostDtTm	Date/Time	8

SDF File Maintenance

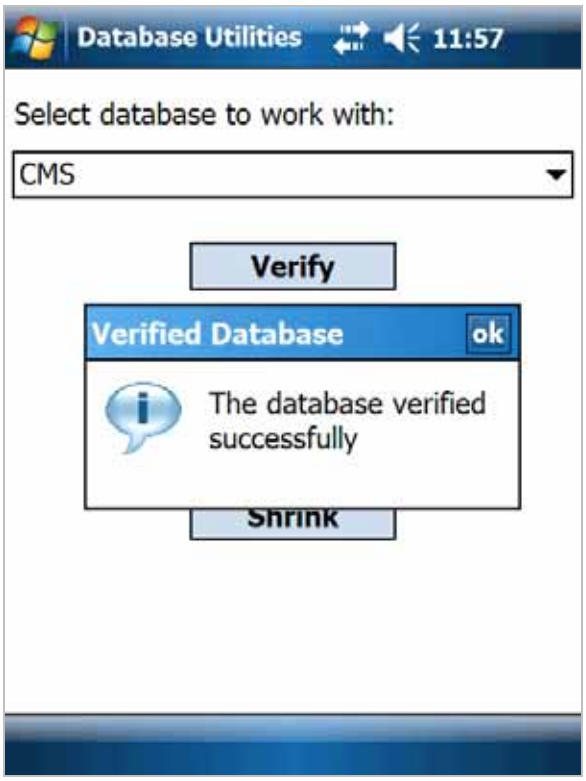
The SDF format database, like all DB systems, can need periodic maintenance to reduce file size, reorganize keys, and, in rare cases, repair corrupted files. We have provided an iPAQ-based utility that will perform these functions on the three GATS SDF files (CMSDB, Survey0, and Survey1). It expects that the SDF files are in the GATS Programs folder. You run this tool by navigating to the GATS Programs folder on the iPAQ and tapping on the GATS_DBUtil program. **Exhibit 6-25** displays the main screen of this utility.

Exhibit 6-25. GATS_DBUtil Start Screen



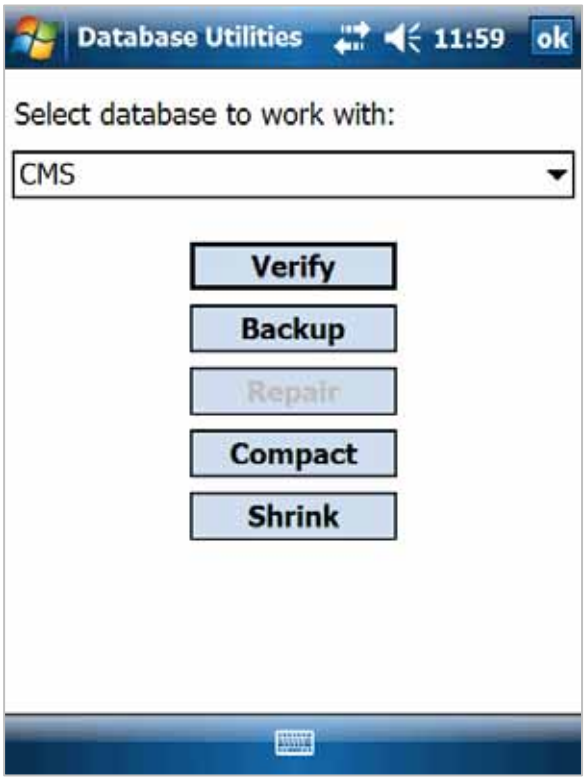
On the startup screen, select one of the three GATS SDF files to work on—the CMS, HH, or IQ. Once this selected you tap **Verify**, as shown in **Exhibit 6-26**.

Exhibit 6-26. GATS_DBUtil after a Verify of CMS



If the file is verified, the other maintenance options are made available; if it is not verified, the Repair option is made available (see below). In the screenshot in **Exhibit 6-27,** the Repair is grayed out since the file was verified. If you need to repair the file, use the backup option before doing the repair as an extra safety measure.

Exhibit 6-27. Gats_DBUtil Options Available After a Successful Verify



7. How to Configure the GSS for a Handheld

This chapter provides the GSS user with the information required to prepare the hardware and software systems to use in data collection. Each subsystem is discussed separately, and users may not use all the subsystems.

7.1 How to Configure a Pocket PC

The GSS is designed to run on a Pocket PC using the Microsoft Windows Mobile 5.0 or 6.0 operating system. The GSS requires a suite of operating system supplements. These are support tools or extensions to the MS Windows Mobile operating system and are free additions available for download from Microsoft. Systems that use languages other than English may need to also load language packs that provide the character set and keyboards (a soft input panel, SIP) for the given language. The required additions are as follows:

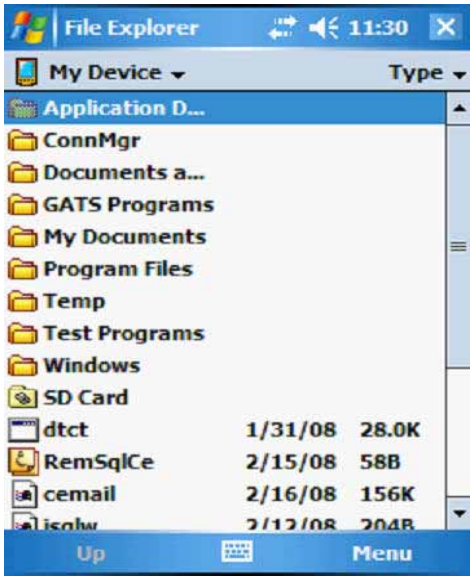
- MS.NET CF 2.0 (.NET compact framework libraries; these are already part of WM 6.0)
- MS SQL Client 2.0 (SQL compact edition—CE)
- MS SQL Server 2005 CE (SQL CE)
- MS Update for 2007 Daylight Savings (patch to fix change in U.S. daylight savings time dates)

These files are available as CAB files from Microsoft and are provided with the GSS distribution. The CAB files are shown below.

Name	Size	Type
DST2007.cab	5 KB	WinZip File
NETCFv2.wm.armv4i.cab	2,185 KB	WinZip File
sqlce30.ppc.wce5.armv4i.CAB	673 KB	WinZip File
sqlce30.repl.ppc.wce5.armv4i.CAB	305 KB	WinZip File

These CAB files must be installed into the operating system. We suggest installation on the device as opposed to external storage. Installation simply requires copying these CAB files to the iPAQ and running them; they are self-installing.

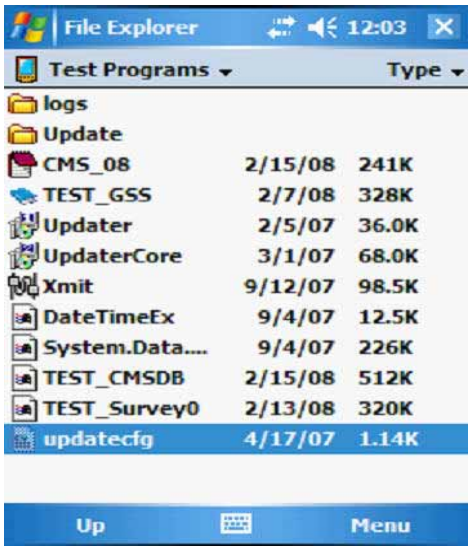
Once the operating system has been augmented with the additions noted above, the user can set up the iPAQ for the GSS tools required. The GSS has a naming convention for file names and locations. It is expected that the user will create a folder in the root level of the iPAQ store named ProjectName Programs. The project name can be any character string used to identify a specific project. It is possible to have multiple projects loaded on the same iPAQ as long as they have different folder names. For this section, we have named the folder TEST and created a folder in the root of the iPAQ store named TEST Programs. The following screenshot shows the listing, using the iPAQ File Explorer utility, of the contents of the root folder. All programs and files required by the GSS should be placed into the Test Programs folder.



The technical support staff will provide software that helps in mastering iPAQ. The mastering tool needs to be configured for the specific hardware model being used, the fonts and languages in use, and any special licensing needs.

7.2 GSS Files

The files that make up the GSS system fall into several categories and are displayed and annotated below.



Subfolders

Logs	Stores log files from transmissions or error reports
Update	Stores files that the Updater program uses

System DLL files

DateTimeEx.dll	dll to allow access to time zone information
System.Data.SQLServerCe.dll	SQL CE support library

Programs

CMS_08.exe	Case Management program
TEST_GSS.exe	GSS Engine program
Updater.exe	Updater program (optional, used only for Web transmission)
UpDaterCore.exe	Updater program startup section (optional, used only for Web transmission)
TEST_Xmit.exe	Transmission program (optional, used only for Web transmission)

Databases

TEST_CMSDB.sdf	The SQL CE database for the CMS
TEST_Survey0.sdf	The SQL CE database that defines Survey0
Updatecfg.xml	Updater configuration file

Typically, the user will modify only the database files unless changes are made to the various programs. The GSS requires at a minimum two database files for a given survey: one database that contains the data tables for driving the CMS and one that defines the survey and provides the input to the GSS engine. If multiple survey forms are used, there will be additional SDF database files, one for each additional survey. For example, three survey forms in a test application would require two additional files: TEST_Survey1.sdf and TEST_Survey2.sdf. The file names must start with the project name convention and an underscore for GSS to find the correct file.

Once these files are copied to the ProjectName folder, in this case TEST Programs, the user can launch the GSS by tapping CMS_08. This step will launch the CMS system and the other programs except the Updater. The GSS expects Updater to be launched by hand when field staff are informed that updates are available. You can create a shortcut for any of these programs and place it in the Windows\StartMenu folder so that they can be launched from the Start menu on the handheld Today screen.

7.3 How to Configure the Case Management System

The CMS has two startup requirements. First, an initialization file in the CMS database (the ProjectName_CMSDB.sdf file) allows the user to set a number of startup parameters. Second, a file or table that provides the source data for cases **must** to be loaded into the CMS. This second requirement depends upon how data are transferred. The following sections describe how to set up the INI table for the CMS. Additional sections detail how to prepare files or tables for the CMS case loading; one section describes the process for using a Project Web site to manage data movement, and another section describes the process for using memory cards to manage data movement.

The CMS program reads startup parameters from a table in the ProjectName_CMSDB.sdf file when it starts. This is the INI table in the database. It a simple table structure with only three fields:

- VariableName
- VariableValue
- Comment

Each row contains a configuration parameter with a unique ID stored in the Variable Name field. The value of the parameter is stored in the Variable Value field, and the Comment field is a text comment that describes the function of the parameter. **Exhibit 7-1** shows the INI parameters and describes their functions.

Case Files

To use the CMS (which manages cases), the user must set up the list of cases for the project that the Field Interviewers will use on their handhelds. The case file setup depends on the method being used to distribute data to the handhelds, either a network-based Web site or memory cards. The setup process is described separately below for each data management method.

Setting Up a Case File Using Network-Based Web Distribution

The case file for Web-based systems consists of a data table in the SQL database that supports the Web operations. Typically, this will be an MS SQL database or an SQL Express database. The Cases table consists of a table with one row in a data table for each “case” that a Field Interviewer works. In a study that uses multiple forms, one row in the case table is needed for each form that the user wishes to track and use to collect data. The SQL case table on the Web site must be populated with cases prior to using the Web site to assign cases to Field Interviewers. The Cases table layout, shown in **Exhibit 7-2**, contains 36 variables that can be broken into the categories listed below. Users can use the variables to hold information that is useful for their study. Users can populate this table with output from their sample development procedures and use SQL tools to insert rows into the table. The fields denoted with an * are required.

ID Variables
- CASEID*
- NumCaseID*
- FIID*
- FormNum*
- ProjectName*
- EVTCODE

Sample Variables
- TRACT_BG
- BLOCK
- HC
- SK
- TYPE
- PERIOD
- YEAR
- WAVE
- ALTERNATIVE
- ALTER_CASID

Locator/Address Variables
- STREETNO*
- STREET*
- APT_NUM
- BOX_NUM
- RURAL_ROUTE
- CITY*
- STATE*
- COUNTY*
- ZIPCODE/PINCODE*

CMS Control Data (not to be changed by users)
- SCRNUM
- ROCBASE
- CREATEDATE
- TRAININGFLAG
- FSID
- ACTIVE
- PostDTTM
- POSTED
- Name
- Phone
- OtherID

Exhibit 7-1. CMS Initialization Table, the INI Table in ProjectName_CMSDB.sdf

Variable Name	Variable Value	Comment
AllowAddressEdit	Yes	Yes or No to allow FI to edit address fields of Cases
AllowCaseTransfer	N	Y or N to allow to transfer cases.
Breakoff	Yes	flag to control breakoff option (Yes/No) Allow = Yes, Not allow = No
DefaultConn	RTI_SERVICES	Default connection for the transmit program
DialPwd	xxxxxx	Dial-up password (used for Xmit)
DialUser	yyyyyy	Dial-up user ID (used for Xmit)
DwellingUnitTable	Yes	Does this project use a Dwelling Unit table (Yes/No)
EvtcodeComplete	200, 400	Result codes that are complete quex
EvtCodeFinal	200, 201, 202, 203, 204, 205, 206, 208, 209, 999, 400, 401, 402, 403, 404, 407, 408, 409	Result codes that are final status codes
EvtCodePending	0, 102, 103, 104, 105, 106, 108, 109, 302, 303, 304, 307, 308, 309, 887	Result codes that are pending status codes
EvtCodeScreened	200	Result code indicating screening complete and one person selected for more data collection
FIID	999998	FI's six digit ID
FIName	Jane Field	Name of Field Interviewer, used as a fill in the Q
GOVID	123456	Property ID or serial number of the handheld
GridColWidths	35, 15, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20	Pixel widths of Variables that are in CMS grid
GridVars	CaseID, FormNum,EVTCODE, Type, StreetNo, Street, Apt_Num, City, State, ZipCode, County, HC, Box_Num, Rural_Route, Tract_BG, Block,SK	Variables that are in CMS grid (do not embed CRs, separate by commas)
HARDWARE	HP210	Type of iPAQ used for survey
InternalCMS	No	Use the internal CMS system?
Lang0Font	Tahoma	Font used for Lang 0
Lang0label	Eng	Language zero menu button label

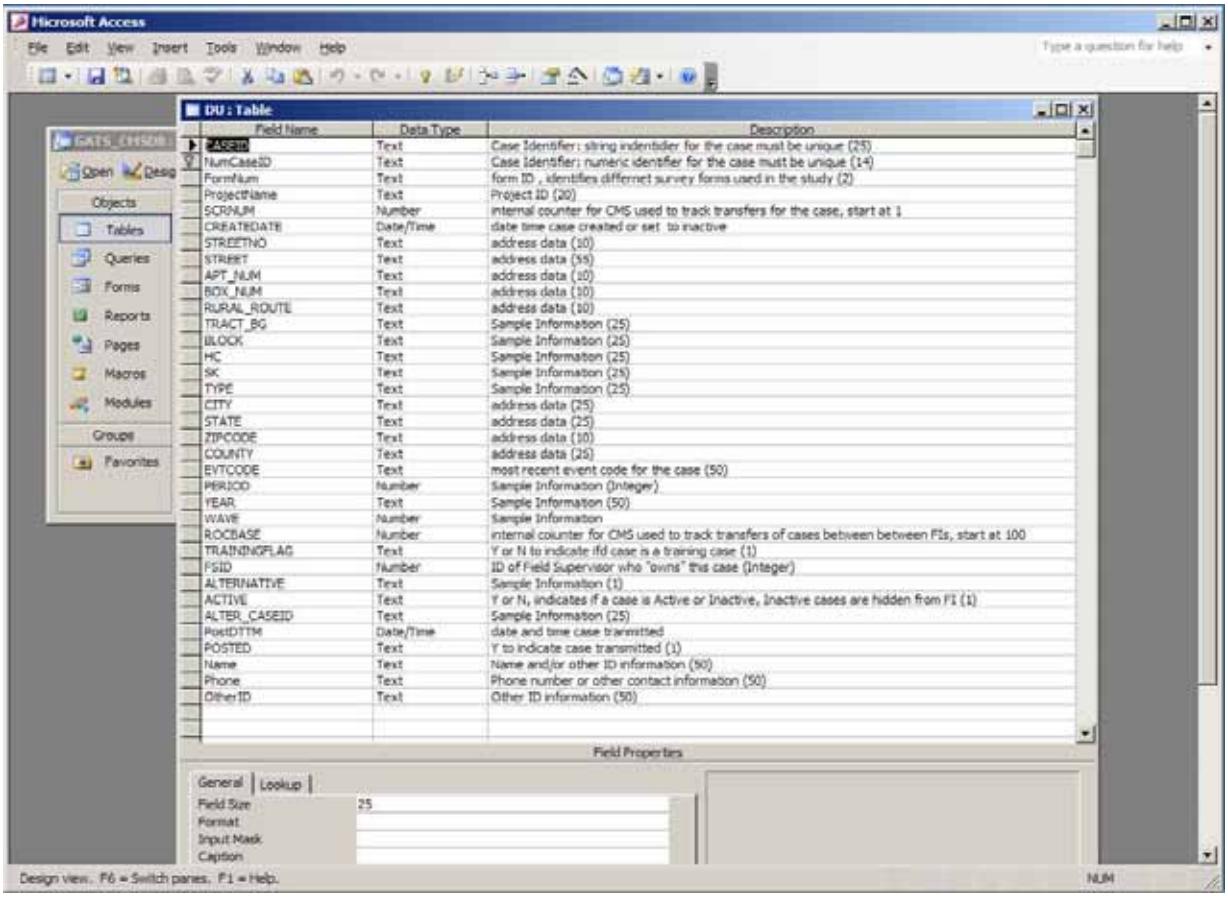
(continued)

Exhibit 7-1. CMS Initialization Table, the INI Table in ProjectName_CMSDB.sdf (continued)

Variable Name	Variable Value	Comment
Language0	ENGLISH	Name of language to use as initial language for Lang0
LockROCok	YES	Lock ROC so cannot add events until HH case is a 200 code
LoginAttempts	0	Used to track the number of failed login attempts
LoginRequired	No	For internal CMS Yes if want to force login
MessageLanguage	ENGLISH	Language for system messages
NumberofSurveys	1	How many different surveys in this project 1 to n
NumLanguages	1	Number of languages used
Password	GATS	User password required to login to GSS
PeriodLimit	3/30/2012|	Used to set test for unlikely login date time
PTEFunction	No	Does this project use the PTE function (Yes/No)
Qversion	Ver. 5.4 Oct 2010	Version of the INI file
QxQOption	Yes	Turn on QxQ option: Yes or No
ShowFormNo	YES	Yes or No to show form No in cases grid
SurveyID	EnglishCore	Unique ID for this Survey
TimeZoneOffset	0	Pseudo Time Zone used by CMS to reset PPC
UserName	a	User Name (used by Xmit)

Exhibit 7-2 provides a description of the data fields, the maximum width of text fields (all text fields are varying character type), and a brief description of the function of that field.

Exhibit 7-2. Data Fields of the Cases Table (CMSDB DU)



Setting Up a Case File for Projects Using Memory Cards to Distribute Cases

Projects that use memory cards to distribute cases to Field Interviewers must build a text version of the Cases table, as described in **Exhibit 7-3**, on a memory card that can be read by the Field Interviewer using the Load Cases feature on the CMS Admin menu. The structure of the text Cases table is a simple tab-delimited format containing all 36 variables listed in **Exhibit 7-3**. Column one must contain the Field Interviewer ID for the Field Interviewer assigned to the case. This file can be built using a tool like Microsoft Excel, with a column for each variable, and then using the Excel export function to build a tab-delimited text file from the spreadsheet. The file name for this file must be GATS_CaseFile.txt, and the file should be placed in the root folder of the memory card on the Field Interviewer's iPAQ. When a Field Interviewer uses the Load Cases option, the GSS will search for a file on the memory card named GATS_CaseFile.txt. If the GSS finds the file, it will load all cases in that file labeled with that interviewer's Field Interviewer ID. It will add cases to a Field Interviewer's iPAQ if the Active field is set to Y. It will remove cases from a Field Interviewer's iPAQ if the Active field is set to N.

Exhibit 7-3. List of Fields for GATS_CASEFILE.txt

Field Name	Contents	Field Name	Contents
FIID	ID of target interview	City	Address info
CaseID	Case ID	State	Address info
NumCase	All numeric case ID must be unique	ZipCode	Address info
FormNum	Set to 0	County	Address info
ProjectName	GATS	EVTCode	Set to 0
SCRNum	Set to 0	Period	Set to 1
CreateDate	Set to date record created	Year	Set to 1
StreetNo	Address info	Wave	Set to 1
Street	Address info	ROC_Base	Leave blank
Apt_Num	Address info	TrainingFlag	Set to N
Box_Num	Address info	FSID	Set to ID of Field Supervisor or 1
Rural_Route	Address info	Alternative	Leave blank
Tract_BG	Sample info	Active	Set to Y to add a case
Block	Sample info		Set to N to remove a case
HC	Sample info	Alter_CaseID	Leave blank
SK	Sample info	PostDtTm	Leave blank
Type	Set to Male or Female or Both	Posted	Leave blank
		Name	Leave blank
		Phone	Leave blank
		OtherID	Leave blank

7.4 How to Configure the GSS

The GSS program, like the CMS, reads in a startup or initialization table at program launch. Many user parameters can be set to start the GSS; these are all stored in the SQLCE SDF file in table INI. The user can set these parameters using any tools that manipulate the tables in the SDF database. Each distinct form used in a given project has its own SDF file and its own INI table. Users most often will use the same parameters across forms. **Exhibit 7-4** details the parameters that are available for the user to set up the GSS. The table shows the default value or recommended value for GATS.

Exhibit 7-4. GSS Initialization Table, the INI Table

Variable Name	Variable Value	Comment
Breakoff	Yes	Flag to control breakoff option (Yes/No)
DETrain	No	Flag to turn on DE Training
DisplayOldData	Yes	If old answers exist show them on back up (Yes/No)
FastForwardOK	Yes	Toggle to turn on or off Fast Forward option (Yes/No)
InternalCMS	No	Use the internal CMS system (Yes/No)
Lang0Font	Tahoma	Font used for Language 0
Lang0label	Eng	Language zero menu button label
Lang1Font	Tahoma	Font used for Language 1
Lang1Label	Ara	Language 1 menu button label
LoginRequired	No	For internal CMS, Yes if want to force login
MessageLanguage	ENGLISH	Language for system messages
NumLanguages	2	Number of languages used
Password	GATS	User password required to login to GSS internal CMS, use only for internal CMS
Qversion	Ver MM DD YYYY	Version of the INI file
QxQOption	Yes	Turn on or off QxQ option (Yes/No)
ROSTERLABEL1	Name|Age| B Mo|B Yr| Gender|Smkr|	Labels for the Roster Columns (0…n)
StartUpLanguage	0	Language to use of Q start up (0 or 1)
TestingRndID	No	Used to generate random IDs if = Yes
TextToCaps	Yes	Convert all text entry to Caps (Yes/No)
UserName	xxxxx	User name for GSS internal CMS, if used
AllowCaseTransfer	Yes	Yes/No to allow case transfer option
AllowAddressEdit	Y	Y/N option to allow editing of case addresses
ShowFormN0	YES	YES/NO option to allow use form number is case grid
SurveyID	XXXXXXX	Text name of Survey must be Unique
NumberofSurveys	2	Number of surveys in GATS 2
TimeZone Offset	0	Not use in GATS default is set to zero
LangSpecsXX	Lang, Font, Size	Used to specify language name, font name and font size for a given language XX = 01 – 25, e.g., MANDARIN,Arial Unicode MS,10
Language0	ENGLISH	Name of Language 0
FiName	Jane Field	Name of FI
FIID	999998	ID # of Field Interviewer
MaxSymbolTableSize	1000	Maximum number of symbols allowed in symbol table
MaxRosterNumber	4	Maximum number of Roster columns
ALLOWNOTEMENUITEM	YES	Option to allow note menu item in GSS for notes on each question

7.5 How to Configure the Transmit Program

The Transmit program needs to have its internal code pointed to the correct Web site or URL. Its other setup parameters can be updated from the Admin menu of the CMS and the startup screens of the program itself.

7.6 How to Configure the Updater Program

The Updater has an XML format configuration file that must be modified to point to the correct Web locations and the assembly names that the user wishes to check. The XML file, updatecfg.xml, is shown below:

```
<?xml version="1.0" encoding="utf-8" ?>
<assemblylist>
      <updateinfo>
          <checkassembly name="CMS_08.EXE" />
          <remoteapp name="CMS_08" />
          <service url="https://nhescms.rti.org/ws_update/agent.asmx"/>
          <version major="0" minor="0" build="0"/>
      </updateinfo>
      <updateinfo>
          <checkassembly name="GATS_GSS.EXE" />
          <remoteapp name="GATS_GSS" />
          <service url="https://nhescms.rti.org/ws_update/agent.asmx"/>
          <version major="0" minor="0" build="0"/>
      </updateinfo>
      <updateinfo>
          <checkassembly name="GATS_XMIT.EXE" />
          <remoteapp name="GATS_XMIT" />
          <service url="https://nhescms.rti.org/ws_update/agent.asmx"/>
          <version major="0" minor="0" build="0"/>
      </updateinfo>
<updateinfo>
<checkassembly name="GATS_PATCH.EXE" />
<remoteapp name="GATS_PATCH" />
<service url="https://nhescms.rti.org/ws_update/agent.asmx"/>
<version major="0" minor="0" build="0"/>
</updateinfo>
<services>
          <address url="http://rcdrhughes/nhes_Web2/ws_update/agent.asmx"/>
<address url="https://nhescmsdev.rti.org/ws_update/agent.asmx"/>
          <address url="https://nhescms.rti.org/ws_update/agent.asmx"/>
      </services>
</assemblylist>
```

7.7 Language Customization

The GSS programs allow the use of multiple languages for survey forms; in addition, all of the program screen text and system messages can be set to a specific language. In the CMSDB.sdf file, a Messages table provides translation of messages to a target language. The default language for GSS is English, but users may translate the English messages to their target language. There are approximately 600 English messages that define the GSS programs. These messages are used for labels, column headings, information messages, and so forth. Each message has a unique ID, a language name, and text. If users

want to develop their own set of messages, they should add additional rows to the Messages table, one row for each English message, with the same message ID but a new language name. **Exhibit 7-5** shows the layout of the Messages table.

Exhibit 7-5. Messages Table

PK	MsgId	LanguageID	Message
5	frmLists001	ENGLISH	You must provide an answer. Please select one line in the list box.
6	frmLists002	ENGLISH	Missing answer.
7	frmLists003	ENGLISH	Don't use this button. Use Next or Back to move from screen to screen.
8	frmLists004	ENGLISH	OK not allowed.
10	frmNumIn001	ENGLISH	You must provide an answer.

The fields are as follows:

- PK: an auto-number field that is the primary key
- MsgID: a unique ID for each message
- Language ID: identifies the language of the message
- Message: the verbatim text of the message

If users want to change the language, they should add a new row for each message. For example, to add a new message for French, the user will start with message "frmLists001" and add a row with the three fields, as shown below:

Frmlists001 FRENCH French text for this message

Typically, users will develop a matching message in the target host language for every English message in the Messages table. GSS looks for messages in the table using the language defined by the INI table parameter, MessageLanguage. If GSS cannot find the message in the language specified by MessageLanguage, then it looks for that message in English.

Note: **Appendix C** contains frequently asked questions and answers. **Appendix D** provides a list of acronyms used in this manual and their definitions. **Appendix E** shows an example mastering process for the iPAQ Model 2490. **Appendix F** shows an example mastering process for the iPAQ Model 210.

Appendix A. GSS Programming Language

This section provides the GSS user with a description of the syntax and usage for the GSS programming language. This language is used in the Qlogic block of the metadata that defines a GSS question or screen. The programming logic is available to allow developers to add flow control, variable creation, evaluate expressions, and call user-supplied subroutines.

General Syntax

The general format of a GSS logic statement is a text sentence ending with a semicolon, for example:

If {HH1}="15" then goto A1;

The sentence above is an If then statement that uses a symbol (the text {HH1} defines a symbol) and a GoTo statement. GSS allows up to 150 sentences in a given logic block for a question. Each sentence should end with a semicolon.

Operators

The GSS If Then statements support the standard mathematical operators allowed in VB.net (<, >, <=, >=, =). If you want to force a numeric comparison instead of a character comparison, code the comparison using a # symbol to force a numeric compare. For example, if you code:

If {V1} = 2 and the symbol {V1} equals 02, then the compare will be false but if you code If {#V1} = 2, then a numeric comparison will be done and the compare will be true.

Symbols

GSS supports the concept of symbols. In a regular programming language, these are called variables. GSS maintains an internal table of symbols called the symbol table. Symbols are added to the symbol table automatically as progress is made through an interview and users can add symbols to the table in logic blocks with a set statement. For example, every time a question is answered, a symbol is created or updated in the symbol table and, hence, is available for users as fill-in question text or in calculations in logic blocks. For example, if you wanted to use a symbol in question text, you would enter the text in the Texts table. If on question A1 you collected a person's first name, then you could use that first name of subsequent questions. The text for a subsequent question might look like this where you use the person's first name, stored in sysmbol {A1}, to personalize the question:

{A1}, can you tell me how old you were on your last birthday?

GSS would look up the value of the symbol {A1} and substitute its current value in the text when the question is presented.

A symbol is defined as a variable name surrounded by curly braces, for example {HH1} or {FirstName}. Symbol names must be less than or equal to 20 characters and must be unique within a questionnaire.

Symbols can have a subscript to indicate the instance level at which they exist. Note that if one is inside loops in GSS, then the variables are at instance levels greater than 0. If the subscript is left off GSS assumes the symbol is at instance level 0.

Statements

The GSS logic parser supports the following statement types:

- If Then
- Set
- GoTo
- Expression evaluation
- Call

If Then Statements

If expression then statement;

This type of statement is a standard If Then programming statement that lets conditional logic be applied. **Table A-1** shows several examples of the syntax for this statement. If the expression evaluates to True, then the statement following the “then” will be executed. If the expression evaluates to False, then the statement after the “then” will not be executed. If there are multiple If Then statements in a logic block, they are executed sequentially unless there is a GoTo statement that is executed, execution of subsequent logic will cease after the first executed GoTo.

Set

The Set statement allows users to create new variables or update the values of variables in the symbol table. The general format is:

Set {V1}=“Harry”;

The above statement creates a symbol V1, if none exists, or updates the existing symbol V1 with the value “Harry.” Symbols can have a subscript to indicate the “instance” level at which they exist. Note that if one is inside loops in GSS, then the variables are at instance levels greater than 0. If the subscript is left off, GSS assumes the symbol is at instance level 0. The syntax to specify a specific symbol noting its instance level is as follows:

Set {V1_0}=“Harry”;

Set {V1_3}=“Harry”;

GoTo

The GoTo statement allows users to skip directly to a new question (QID). The syntax of the GoTo is:

GoTo XXX;

where XXX is the valid name of a QID in the current questionnaire. GSS does not check the validity of the QID until run time so if the QID does not exist it will cause a run time error on the handheld.

BackUpTo

The BackUpTo statement allows users to back up to an old question (QID). The syntax of the BackUpTo is:

```
BackUpTo XXX;
```

where XXX is the valid name of a QID in the current questionnaire. GSS does not check the validity of the QID until run time so if the QID does not exist it will cause a run time error on the handheld. The BackUpTo statement allows users to code flow control that will back up correctly and invalidate data as it backs up over questions. For example if you had answered questions Q1, Q2, Q3, and Q4 and then issued, on Q4, a BackUpTo Q1 command then the data for Q3,Q2, and Q1 would be flagged as invalid showing that it had been backed over.

Expression

A GSS expression is a standard programming Visual Basic style expression. Users can use built-in functions from Visual Basic like RND() or Len() or regular math and logic expressions. Expressions can be used in If Then statement or on the right of set statements. Examples of expression can be seen in **Table A-1**.

The lines below present addition examples of logic statements and ones using functions.

```
{V1}=rnd()*100 + (30/4);

{Tdy}=today();

{TimeNow}=Now();

{X}=len(Now());

if {B13a} = "1" and ({#B13b} < 1 or {#B13b} > 120) then goto B13c;

if {A1}=“Joe” then BackUpTo HH5;
```

Exhibit A-1. Examples of Statements

Statement	Examples
BackUpTo If Then	BackUpTo HH1;
Set	Set {V21}=55; Set {V1}=”Name”; set {Agency}="Statistical Agency"; set {CountryName}="Country"; set {CountryExamples_0}="cigarettes, cigars, pipes"; set {Fill1_0}="next oldest"
Goto	Goto HH1;

Expressions	if {#v1} >= 75 then goto Question1;
	if {#HH4b} >= 15 and {#HH4b} <= 17 then goto HH4c;
	if {#HH4b} < 15 then goto TooYoung;
	if {HH4c} = "77" or {HH4c} = "99" or {HH4cYear} = "?" or {HH4cYear} = "!" or {HH4cYear} = "7777" or {HH4cYear} = "9999" then goto HH4d;
	if {#calcyears} < 15 then goto ValidateAge;
	if {HH5Flag} = "1" then set {EventCode} = "200";
	if {NoEflag} = "1" then set {EventCode} = "201";
	if {HH5Flag} = "1" then set {EventComment} = "Screener Complete";
	if {NoEflag} = "1" then set {EventComment} = "Screener Complete No Eligibles";
	call Code_Event;
	if true then set {C14}="0";

Call	Call SampleSelect; If {A32}=“YES” then Call UserCode; call ValidateBDay;

The **BackUpTo** statement forces GSS to backup (correctly flagging invalid data as it backs up) to the target screen provided as the one parameter for BackUpTo. The **Goto** function forces GSS to jump directly to the screen provided as a target, it is not recommend that you use GoTo to jump backwards as it may leave data orphaned as it jumps back over completed questions, jumping forward is fine. The **Set** function causes a symbol to be created and have a value stored in it that can be used later in the program using the {} conventions to call back variables from the symbol store.

Call

The call statement allows users to call user written or existing system installed subroutines that can create custom code. If users wish to create callable subroutines, they must code the new routines and generate a new compiled version of GSS with the callable routines.

Visual Basic.net Functions

Several VB.net functions can be used inside the programming language. These are standard MS functions and their documentation is covered in any VB.net manual. The list of functions is shown in **Exhibit A-2** and the calling sequence is shown in the code listing presented in **Exhibit A-3**. The name of

the function is case sensitive as shown in the table below. Additional functions can be added if needed by extending the class EvalFunctions.

Exhibit A-2. Functions Available in Qlogic Sentences

String Functions	Math Functions	Miscellaneous Functions	Date/Time Functions
Instr	Sin	Format	DatePart
Rnd	Rnd		Now
Right	Random		[Date]
Mid	Round		Today
Mid1	ConvInt		
Len	Int		
Trim			
Ucase			
LCase			
Left			

Exhibit A-3. VB.net Function Calling Specifications (Page 1 of 3)

```
Public Class EvalFunctions

 ' FUNCTION HERE CAN BE USED IN ANY EVALUATION FORMULA
 ' PARAMETERS AND RETURN VALUES CAN BE
 ' DOUBLE (do not use integer !)
 ' DATETIME
 ' BOOLEAN
 ' Jan 2008 Modified for Messages Table (RBK)

 Function GetSQL(ByVal caseid As String, ByVal qid As String)
 ' Get data from responses table in CMS data base
 ' Get ONLY valid data
 '
 ' always get it from CMS database
 '
 '
 ' added to routines August 2007


 Dim DR As SqlCeDataReader
 Dim RetVal As String = ""
 Dim sqlCmd2 As New SqlCeCommand("Select * from Responses where valid=1 and CaseID =
'" & caseid & "' and QID = '" & Qid & "'", CMSConn)

 Try
 DR = sqlCmd2.ExecuteReader

 While DR.Read
 ' will get the latest version of the answer that is valid
 'MsgBox(DR.Item("Answer"), MsgBoxStyle.Information, "Values GETRESPONSEDATA")
 RetVal = DR.Item("Answer")
```

(continued)

Exhibit A-3. VB.net Function Calling Specifications (Page 2 of 3)

```
 End While
 Catch ex As Exception
 '   MsgBox("Error   in   Select   from   Get   Case   data   from   Responses   table",
MsgBoxStyle.Critical, "getResponseCaseDataValid:SQL Error")
 MsgBox(GetMessage("EvalFunctions001"),                          MsgBoxStyle.Critical,
GetMessage("EvalFunctions002"))
 Application.Exit()
 End Try

 Return RetVal
 End Function

 Function Instr(ByVal Start As Double, ByVal S1 As String, ByVal S2 As String) As
String
 Return Microsoft.VisualBasic.InStr(CInt(Start), S1, S2)
 End Function


 Function Left(ByVal S1 As String, ByVal Leng As Double) As String
 Return Microsoft.VisualBasic.Left(S1, CInt(Leng))
 End Function

 Function Right(ByVal S1 As String, ByVal Leng As Double) As String
 Return Microsoft.VisualBasic.Right(S1, CInt(Leng))
 End Function

 Function DatePart(ByVal Part As String, ByVal DateVal As String) As Double
 Dim NewDate As Date
 NewDate = CDate(DateVal)
 Return Microsoft.VisualBasic.DatePart(Part, NewDate)
 End Function

 Function Sin(ByVal v As Double) As Double
 Return Math.Sin(v)
 End Function

 Function Now() As DateTime
 Return Microsoft.VisualBasic.Now
 End Function

 Function Today() As DateTime
 Return Microsoft.VisualBasic.Today
 End Function

 Function Rnd() As Integer
 Microsoft.VisualBasic.Randomize()
 Return CInt(Microsoft.VisualBasic.Rnd() * 100)
 End Function

 Function Random() As Integer
 Microsoft.VisualBasic.Randomize()
 Return CInt(Microsoft.VisualBasic.Rnd() * 100)
 End Function


 Function Mid(ByVal str As String, ByVal index As Double) As String
 Return Microsoft.VisualBasic.Mid(str, CInt(index))
 End Function
```

(continued)

Exhibit A-3. VB.net Function Calling Specifications (Page 3 of 3)

```
 Function Mid1(ByVal str As String, ByVal index As Double, ByVal len As Double) As
String
 Return Microsoft.VisualBasic.Mid(str, CInt(index), CInt(len))
 End Function

 Function Len(ByVal str As String) As Double
 Return Microsoft.VisualBasic.Len(str)
 End Function

 Function Trim(ByVal str As String) As String
 Return Microsoft.VisualBasic.Trim(str)
 End Function

 Function Format(ByVal value As Object, ByVal style As String) As String
 Return Microsoft.VisualBasic.Format(value, style)
 End Function

 Function UCase(ByVal value As String) As String
 Return Microsoft.VisualBasic.UCase(value)
 End Function

 Function LCase(ByVal value As String) As String
 Return Microsoft.VisualBasic.LCase(value)
 End Function

 Function [Date](ByVal year As Double, ByVal month As Double, ByVal day As Double) As
DateTime
 Return New Date(CInt(year), CInt(month), CInt(day))
 End Function

 Function Int(ByVal value As Double) As Double
 Return Microsoft.VisualBasic.Int(value)
 End Function

 Function Round(ByVal value As Double) As Double
 Return Math.Round(value)
 End Function

 Function ConvInt(ByVal value As String) As Double
 Return CDbl(value)
 End Function

End Class
```

Appendix B. Review of the Main Data Tables

This section of this document aims to provide the information and detail needed to work with the GSS data tables once data has been collected in a survey. In GSS all the data collected during survey work are stored in an SDF database in tables. These tables are stored in the GATS_CMSDB.sdf. In GATS, the data from both survey forms are stored within the same table and are distinguished by their different CaseIDs and/or Form Numbers. In the GSS three main data tables store data during the course of a survey (others play a minor role (address logs and case notes)). The three main tables are:

- The **DU** table (see **Exhibit B-1**). This table contains one row for each case that is to be worked. These rows are loaded into the iPAQ as field work progresses to provide the cases that an Field Interviewer is to work; in other words, to define the survey sample. In GATS two cases or data collection activities will be worked for each dwelling unit (DU): one activity for the screener and, if the screener is successful in yielding a person to interview, then a second activity, the individual interview. The DU records contain case identification information, sample information, and geographic locator information (addresses or other information to identify and locate the household). It also holds a copy of the most recent result or status code.
- The **Responses** table (see **Exhibit B-2**). This table contains the data that Field Interviewers enter as they step through a GSS questionnaire. There is one row in the Responses table for each question asked in a GSS questionnaire. Every time the Field Interviewer presses the **Next** button, a new row is written to the Responses table with the data collected on that screen. The Responses table also stores internal or system variables that GSS or the GSS programmer output during an interview.
- The **DUEvent** table (see **Exhibit B-3**). This table contains the status or result codes collected during the course of working a case. These codes may be generated automatically during or interview or added by hand by the Field Interviewer. For more information about the generation and management of event data, see the ***GATS Field Interviewer Manual*** and Field Supervisor manual.

The specifications below detail the variables or columns of the three data tables. The DU table has one row for every case where a case is a given data collection activity, either a screener questionnaire or an IQ in GATS. Its layout and the definitions of the columns are shown in **Exhibit B-1**.

Exhibit B-1. DU Table—Table of Case-Level Records for Each Unique Data Collection Activity

Column Name	Definition	Type	Size in Bytes	Notes
CaseID	Unique ID for the case, used in all case tables	NVarChar	25	Must be unique within the entire survey sample
NumCaseID	Alternate ID or 2ndary for the case	NVarChar	14	Not needed but sometime used for addition ID information
FormNum	Questionnaire question ID	NVarChar	2	In GATS 0=Screener 1=IQ
ProjectName	Raw data answer to question QID	NVarChar	20	Used to identify the project within and between countries
EvtCode	Most recent result code for this case	NVarChar	50	
ScrNum	Number of times that this case has been reassigned	SmallInt	2	Only used with Web-based case assignment
CreateDate	Date case record/row created	Date/Time	8	
StreetNo	Street number of dwelling unit or survey location	NVarChar	10	**All rows shaded gray are locator information that can be adapted country by country**
Street	Name of street	NVarChar	55	
Apt_Num	Field Interviewer ID number	NVarChar	10	
Box_Num	Box number of dwelling unit	NVarChar	10	
Rural_Route	Rural route number of dwelling unit	NVarChar	10	
City	Name of city	NVarChar	25	
State	Name of state	NVarChar	25	
ZipCode	Zip code	NVarChar	10	
County	Name of county	NVarChar	25	
Tract_BG	Sample Information	NVarChar	25	All sample information rows can be adapted country by country
Block	Sample Information	NVarChar	25	
HC	Sample Information	NVarChar	25	
SK	Sample Information	NVarChar	25	
Type	Sample Information	NVarChar	25	
Period	Sample Information	SmallInt	2	
Year	Sample Information	NVarChar	50	
Wave	Sample Information	SmallInt	2	
ROCBase	Base code for event data	SmallInt	2	Only used by Web case assignment

(continued)

Exhibit B-1. DU Table—Table of Case-Level Records for Each Unique Data Collection Activity (continued)

Column Name	Definition	Type	Size in Bytes	Notes
TrainingFlag	Flag to indicate a training case	NVarChar	1	Y=Training case; N=live case
FSID	ID of Field Supervisor who owns this case	SmallInt	2	Only used by Web case assignment and transmission
Alternative	Flag to indicate if this case is an alternative case in a pair of cases	NVarChar	1	Not used in GATS
Active	Flag to indicate if the case is active or not	NVarChar	1	Y or N; if set to N, case will not display on an FI's iPAQ
Alter_CaseID	ID of paired case	NVarChar	25	Not used in GATS
PostDtTm	Timestamp when case was xmited	Date/Time	8	Only used by Web case assignment and transmission
Posted	Flag to indicate row was transmitted	NVarChar	1	Only used by Web case assignment and transmission
Name	ID information of people or person within HH	NVarChar	50	Can be used to store information collected during screening or interview to display in case info
Phone	ID information of people or person within HH	NVarChar	50	Can be used to store information collected during screening or interview to display in case info
OtherID	ID information of people or person within HH	NVarChar	50	Can be used to store information collected during screening or interview to display in case info

The Responses table (see **Exhibit B-2**), contains question-level data and internal variables generated by GSS or the GSS programmer during an interview. In general, it contains one row of information for each question asked during an interview. Every time an interviewer presses the **Next** button on a screen, a row of data is written to the Responses table. In GSS only one data item is stored per screen; if there are multiple data items on a screen, they are concatenated together, separated by the "|" symbol, and stored in the Answer column.

Exhibit B-2. Responses Table—Table to Store Respondent Answers

Column Name	Definition	Type	Size in Bytes	Required Data	Notes
CaseID	Unique ID for the case	NVarChar	25	Yes	Must be unique within the entire survey sample, same ID as in DU table
NumCaseID	Alternate ID or 2ndary for the case	NVarChar	14	No	Must be unique within the entire survey sample, same ID as in DU table
QID	Questionnaire question ID	NVarChar	20	Yes	Must be unique within a questionnaire
Answer	Raw data answer to question QID	NVarChar	255	Yes	Note that GSS collects only one data item per screen
Instance	Indicates level of looping if within a loop in the questionnaire	Long Int	8	Yes	Range 0 to n, where 0 indicates not in a loop and >0 indicates in a loop
Lang	Code indicating language used on this screen	Int	2	Yes	In the range 0-25
DtTm	Time stamp	Date/Time	8	Yes	Standard MS date time
Version	Version stamp of the questionnaire	NVarChar	50	Yes	Version info comes from text associated with GSS START screen
UserID	Field Interviewer ID number	NVarChar	20	Yes	Must be unique over FIs
Prikey	Auto number field used to number rows	Long Int	4	Yes	Numbers rows uniquely within an iPAQ
Valid	Flag to indicate if data field has been backed over or invalidated	Boolean	1	Yes	True or false
OnHold	Yes/No flag indicating row on hold for transmission	NVarChar	1	No	Used only for Web-based transmission
Posted	Yes/No flag indicating xmit status	NVarChar	1	No	Used only for Web-based transmission
PostDtTm	Time stamp for xmit	Date/Time	8	No	Used only for Web-based transmission
TZ	Time Zone indicators	SmallInt	2	No	Not used in GATS
TZ1	Time Zone indicators	SmallInt	2	No	Not used in GATS

The DUEvent table (see **Exhibit B-3**), contains a row for each event that is recorded for a given case. These events may be generated by the GSS questionnaire or entered by hand by the Field Interviewer. The event table allows one to track and monitor the status of a case as Field Interviewer works it.

Exhibit B-3. DUEvent Table—Table to Record Status Codes for Data Collection Activities or Events

Column	Definition	Type	Size in Bytes	Required Data	Notes
CaseID	Unique ID for the case, linked to DU and Responses table	NVarChar	25	Yes	Must be unique within the entire survey sample, same ID as in DU table and responses table
NumCaseID	Alternate ID or 2ndary for the case	NVarChar	14	Yes	Must be unique within the entire survey sample, same ID as in DU table
Createdate	Date time event created	Date/Time	8	Yes	
Cont_No	Contact number, sequential count of events for this CaseID	SmallInt	2	Yes	
ScrNum	Number of times this CaseID has been reassigned	SmallInt	2	Yes	Starts at 0
EVENTDATE	Date of event as entered by Field Interviewer	Date/Time	8	Yes	Standard MS date time variable
RESULT	Result code	NVarChar	50	Yes	
FIID	ID of interviewer	NVarChar	8	Yes	ID of interviewer
Comments	Free text for Field Interviewer comments	Ntext	16	Yes	Free text for Field Interviewer comments
HISPBOX	Not used in GATS	SmallInt	2	Yes	Not used in GATS
MODIFYDTTM	Time stamp of last mod to an event	Date/Time	8	Yes	Standard MS date time variable
AUTO_EVENT	Yes/No flag used to indicate if event is system generated	NVarChar	1	No	Y=system generated event, N=event entered by Field Interviewer
NULLIFIED	Yes/No flag event has been nullified	NVarChar	1	No	Used only for Web-based transmission
AUTH_CODE	For events that require an authorization code it is stored here if used	NvarChar	5	No	Generally a letter followed by 4 digits, e.g. a1234
RECVDTTM	Time stamp for receipt	Date/Time	8	No	Used only for Web-based transmission
PostDTTM	Time stamp for posting	Date/Time	8	No	Used only for Web-based transmission
POSTED	Yes/No flag to indicate event has been transmitted; Y=transmitted	NVarChar	1	No	Used only for Web-based transmission
TZ	Time Zone indicators	SmallInt	2	No	Not used in GATS
TZ1	Time Zone indicators	SmallInt	2	No	Not used in GATS

Exhibits B-4 and **B-5** show sample data for the DUEvent and Responses tables discussed above. **Exhibit B-4** shows sample response data for a GATS HH questionnaire (CaseID 100027-00) and a portion of the corresponding IQ questionnaire (CaseID 100027-01). Most data rows represent the answer to a given GATS question and the QID matches the questionnaire specification for the question name, for example QIDs HH1, HH2, HH3, etc. In the Reponses table, note that each interview starts with a QID=APPSTART row and ends with a QID=SURVEYEXIT row. These rows mark the beginning and end of an interview and are generated automatically when GSS starts and ends an interview. Also note that there are rows that contain items that are not specific questionnaire items. These rows represent GSS internal variables that capture data that are GSS related but not direct answers to questions. For example, the variables/rows selectee, sage, sgender, selename, and randomnum are all variables that come out of the selection process of a household member for interview that identify the roster row, age, gender, name, and random number associated with the selected person in the roster. Langcalc and Langspoken are variables that indicate the percentages of the screens that were in language 0 and what language (0 or 1) was used for >50% of the screens. In addition, note the field labeled Valid. It contains the value FASLE when a variable item has been backed or and, hence, superseded, otherwise it is TRUE.

Exhibit B-5 contains a sample of GSS event data. Here each row represents one event associated with a specific case. In this table we can see that for CaseID 100027-00 we have an event indicating the screener was completed on 5/25/2008 13:38 and another event indicating the IQ was completed for CaseID 100027-01 on 5/25/2008 14:06. Both of these events are auto events indicating that the event was generated as part of the GSS interview process.

Exhibit B-4. Sample Responses Data

CaseID	NumCase	QID	Answer	Instance	Lang	DtTm	Version	UserId	PriKey	Valid	OnHold	Posted	PostDtTm	TZ	TZ1
100027-00	27-00	APPSTART		0	0	5/25/2008 13:35:12	HQ:V 1.11 .05.22	999002	4286	TRUE	N	N			
100027-00	27-00	INTRO1		0	1	5/25/2008 13:35:18	HQ:V 1.11 .05.22	999002	4287	TRUE	N	N			
100027-00	27-00	HH1	5	0	1	5/25/2008 13:35:27	HQ:V 1.11 .05.22	999002	4288	FALSE	N	N			
100027-00	27-00	HH1	4	0	1	5/25/2008 13:36:01	HQ:V 1.11 .05.22	999002	4289	TRUE	N	N			
100027-00	27-00	HH2	2	0	1	5/25/2008 13:36:44	HQ:V 1.11 .05.22	999002	4290	TRUE	N	N			
100027-00	27-00	HH3	1	0	1	5/25/2008 13:36:55	HQ:V 1.11 .05.22	999002	4291	TRUE	N	N			
100027-00	27-00	HH4		0	1	5/25/2008 13:37:04	HQ:V 1.11 .05.22	999002	4292	TRUE	N	N			
100027-00	27-00	HH4A	Person 1	1	1	5/25/2008 13:37:40	HQ:V 1.11 .05.22	999002	4293	TRUE	N	N			
100027-00	27-00	HH4B	31	1	1	5/25/2008 13:38:25	HQ:V 1.11 .05.22	999002	4294	TRUE	N	N			
100027-00	27-00	HH4D	1	1	1	5/25/2008 13:38:28	HQ:V 1.11 .05.22	999002	4295	TRUE	N	N			
100027-00	27-00	HH4E	2	1	1	5/25/2008 13:38:35	HQ:V 1.11 .05.22	999002	4296	TRUE	N	N			
100027-00	27-00	HH4F	1	1	1	5/25/2008 13:38:37	HQ:V 1.11 .05.22	999002	4297	TRUE	N	N			
100027-00	27-00	EDITROSTERINTRO	2	0	1	5/25/2008 13:38:39	HQ:V 1.11 .05.22	999002	4298	TRUE	N	N			
100027-00	27-00	SELECTEE	1	0	1	5/25/2008 13:38:40	HQ:V 1.11 .05.22	999002	4299	TRUE	N	N			
100027-00	27-00	SAGE	31	0	1	5/25/2008 13:38:40	HQ:V 1.11 .05.22	999002	4300	TRUE	N	N			
100027-00	27-00	SGENDER	MALE	0	1	5/25/2008 13:38:40	HQ:V 1.11 .05.22	999002	4301	TRUE	N	N			
100027-00	27-00	SELNAME	Person 1	0	1	5/25/2008 13:38:40	HQ:V 1.11 .05.22	999002	4302	TRUE	N	N			
100027-00	27-00	RANDOMNUM	0.5386011	0	1	5/25/2008 13:38:40	HQ:V 1.11 .05.22	999002	4303	TRUE	N	N			
100027-00	27-00	HH5		0	1	5/25/2008 13:38:43	HQ:V 1.11 .05.22	999002	4304	TRUE	N	N			
100027-00	27-00	LANGCALC	6	0	1	5/25/2008 13:38:45	HQ:V 1.11 .05.22	999002	4305	TRUE	N	N			
100027-00	27-00	LANGSPOKEN	1	0	1	5/25/2008 13:38:45	HQ:V 1.11 .05.22	999002	4306	TRUE	N	N			
100027-00	27-00	SURVEYEXIT		0	1	5/25/2008 13:38:45	HQ:V 1.11 .05.22	999002	4307	TRUE	N	N			
100027-01	27-01	APPSTART		0	0	5/25/2008 13:39:11	GATS IQ VERSIO	999002	4308	TRUE	N	N			
100027-01	27-01	CONSENTINDIA		0	0	5/25/2008 13:39:16	GATS IQ VERSIO	999002	4309	TRUE	N	N			
100027-01	27-01	CREAD	1	0	0	5/25/2008 13:39:19	GATS IQ VERSIO	999002	4310	TRUE	N	N			
100027-01	27-01	COBTAINED	1	0	0	5/25/2008 13:39:20	GATS IQ VERSIO	999002	4311	TRUE	N	N			
100027-01	27-01	INTLANG	2	0	0	5/25/2008 13:39:22	GATS IQ VERSIO	999002	4312	TRUE	N	N			
100027-01	27-01	A00		0	1	5/25/2008 13:39:24	GATS IQ VERSIO	999002	4313	TRUE	N	N			
100027-01	27-01	A01	1	0	1	5/25/2008 13:39:26	GATS IQ VERSIO	999002	4314	TRUE	N	N			
100027-01	27-01	A02A	5	0	1	5/25/2008 13:40:49	GATS IQ VERSIO	999002	4315	TRUE	N	N			
100027-01	27-01	A02B	1977	0	1	5/25/2008 13:41:19	GATS IQ VERSIO	999002	4316	TRUE	N	N			
100027-01	27-01	A04	1	0	1	5/25/2008 13:41:25	GATS IQ VERSIO	999002	4317	TRUE	N	N			
100027-01	27-01	A05	5	0	1	5/25/2008 13:41:44	GATS IQ VERSIO	999002	4318	TRUE	N	N			
100027-01	27-01	D14	2	0	1	5/25/2008 13:47:51	GATS IQ VERSIO	999002	4319	TRUE	N	N			
100027-01	27-01	D16	4	0	1	5/25/2008 13:49:23	GATS IQ VERSIO	999002	4320	TRUE	N	N			
100027-01	27-01	EE01	3	0	1	5/25/2008 13:49:44	GATS IQ VERSIO	999002	4321	TRUE	N	N			
100027-01	27-01	E04	1	0	1	5/25/2008 13:49:50	GATS IQ VERSIO	999002	4322	TRUE	N	N			
100027-01	27-01	E05	3	0	1	5/25/2008 13:49:57	GATS IQ VERSIO	999002	4323	TRUE	N	N			
100027-01	27-01	E07	1	0	1	5/25/2008 13:50:24	GATS IQ VERSIO	999002	4324	TRUE	N	N			
100027-01	27-01	E08	1	0	1	5/25/2008 13:50:31	GATS IQ VERSIO	999002	4325	TRUE	N	N			
100027-01	27-01	E09	2	0	1	5/25/2008 13:50:48	GATS IQ VERSIO	999002	4326	TRUE	N	N			
100027-01	27-01	E11	2	0	1	5/25/2008 13:50:59	GATS IQ VERSIO	999002	4327	TRUE	N	N			
100027-01	27-01	E13	2	0	1	5/25/2008 13:51:09	GATS IQ VERSIO	999002	4328	TRUE	N	N			
100027-01	27-01	E15	2	0	1	5/25/2008 13:51:21	GATS IQ VERSIO	999002	4329	TRUE	N	N			
100027-01	27-01	E17	1	0	1	5/25/2008 13:51:41	GATS IQ VERSIO	999002	4330	TRUE	N	N			
100027-01	27-01	FB01A	3	0	1	5/25/2008 13:52:07	GATS IQ VERSIO	999002	4331	TRUE	N	N			
100027-01	27-01	FB01B	1	0	1	5/25/2008 13:52:13	GATS IQ VERSIO	999002	4332	TRUE	N	N			
100027-01	27-01	FB01C	xxxx	0	1	5/25/2008 13:52:30	GATS IQ VERSIO	999002	4333	TRUE	N	N			
100027-01	27-01	FB02	5	0	1	5/25/2008 13:52:41	GATS IQ VERSIO	999002	4334	TRUE	N	N			
100027-01	27-01	FB03	4	0	1	5/25/2008 13:52:48	GATS IQ VERSIO	999002	4335	TRUE	N	N			
100027-01	27-01	FB04	3	0	1	5/25/2008 13:53:08	GATS IQ VERSIO	999002	4336	TRUE	N	N			
100027-01	27-01	G01INTRO		0	1	5/25/2008 13:53:12	GATS IQ VERSIO	999002	4337	TRUE	N	N			
100027-01	27-01	G01A	2	0	1	5/25/2008 13:53:26	GATS IQ VERSIO	999002	4338	TRUE	N	N			
100027-01	27-01	G01B	1	0	1	5/25/2008 13:53:35	GATS IQ VERSIO	999002	4339	TRUE	N	N			
100027-01	27-01	G01C	2	0	1	5/25/2008 13:53:41	GATS IQ VERSIO	999002	4340	FALSE	N	N			
100027-01	27-01	G01C	7	0	1	5/25/2008 13:53:44	GATS IQ VERSIO	999002	4341	TRUE	N	N			
100027-01	27-01	G01D	2	0	1	5/25/2008 13:53:54	GATS IQ VERSIO	999002	4342	TRUE	N	N			
100027-01	27-01	G01E	2	0	1	5/25/2008 13:54:15	GATS IQ VERSIO	999002	4343	FALSE	N	N			
100027-01	27-01	G01E	1	0	1	5/25/2008 13:54:25	GATS IQ VERSIO	999002	4344	TRUE	N	N			
100027-01	27-01	G01E1	mouth publicity	0	1	5/25/2008 13:54:58	GATS IQ VERSIO	999002	4345	TRUE	N	N			
100027-01	27-01	G02	2	0	1	5/25/2008 13:55:19	GATS IQ VERSIO	999002	4346	TRUE	N	N			
100027-01	27-01	G04A	2	0	1	5/25/2008 13:55:52	GATS IQ VERSIO	999002	4347	TRUE	N	N			
100027-01	27-01	G04B	2	0	1	5/25/2008 13:56:16	GATS IQ VERSIO	999002	4348	TRUE	N	N			
100027-01	27-01	G04C	2	0	1	5/25/2008 13:56:25	GATS IQ VERSIO	999002	4349	FALSE	N	N			
100027-01	27-01	G04C	1	0	1	5/25/2008 13:56:30	GATS IQ VERSIO	999002	4350	FALSE	N	N			
100027-01	27-01	G04C	2	0	1	5/25/2008 13:56:48	GATS IQ VERSIO	999002	4351	TRUE	N	N			
100027-01	27-01	G04D	9	0	1	5/25/2008 13:57:01	GATS IQ VERSIO	999002	4352	TRUE	N	N			
100027-01	27-01	G04E	2	0	1	5/25/2008 13:57:13	GATS IQ VERSIO	999002	4353	TRUE	N	N			
100027-01	27-01	G04F	1	0	1	5/25/2008 13:57:40	GATS IQ VERSIO	999002	4354	TRUE	N	N			
100027-01	27-01	G04G	2	0	1	5/25/2008 13:58:17	GATS IQ VERSIO	999002	4355	TRUE	N	N			
100027-01	27-01	G04H	2	0	1	5/25/2008 13:58:19	GATS IQ VERSIO	999002	4356	TRUE	N	N			
100027-01	27-01	G04I	2	0	1	5/25/2008 13:58:44	GATS IQ VERSIO	999002	4357	TRUE	N	N			
100027-01	27-01	G04J	1	0	1	5/25/2008 13:58:55	GATS IQ VERSIO	999002	4358	TRUE	N	N			
100027-01	27-01	G04K	2	0	1	5/25/2008 13:59:12	GATS IQ VERSIO	999002	4359	TRUE	N	N			

Exhibit B-5. Sample Event Data

CASEID	Num CaseID	CREATE DATE	CONT_NO	SCRNUM	EVENTDATE	RESULT	FIID	COMMENTS	HISPBOX	MODIFYDTTM	AUTO_EVENT	NULLIFIED	AUTH_CODE	RECVDTTM
100101-00	0	5/15/2008 13:36:47	0	0		0	999001							
100024-01	0	5/25/2008 16:59:37	0	0		0	999002							
100022-00	0	5/25/2008 12:30:18	1	0	5/25/2008 12:30:18	200	999002	Complete Screener 1		5/25/2008 12:30:18	Y	N		
100022-01	0	5/25/2008 12:52:51	1	0	5/25/2008 12:52:51	400	999002	Complete IQ		5/25/2008 12:52:51	Y	N		
100024-00	0	5/25/2008 13:03:22	1	0	5/25/2008 13:03:22	200	999002	Complete Screener 1		5/25/2008 13:03:22	Y	N		
100025-00	0	5/25/2008 13:14:55	1	0	5/25/2008 13:14:55	200	999002	Complete Screener 1		5/25/2008 13:14:55	Y	N		
100026-00	0	5/25/2008 13:31:53	1	0	5/25/2008 13:31:53	200	999002	Complete Screener 1		5/25/2008 13:31:53	Y	N		
100027-00	0	5/25/2008 13:38:45	1	0	5/25/2008 13:38:45	200	999002	Complete Screener 1		5/25/2008 13:38:45	Y	N		
100027-01	0	5/25/2008 14:06:39	1	0	5/25/2008 14:06:39	400	999002	Complete IQ		5/25/2008 14:06:39	Y	N		
100026-01	0	5/25/2008 14:23:55	1	0	5/25/2008 14:23:55	400	999002	Complete IQ		5/25/2008 14:23:55	Y	N		
100033-00	0	5/25/2008 16:59:39	0	0		0	999003							
100035-00	0	5/25/2008 12:31:22	1	0	5/25/2008 12:31:22	200	999003	Complete Screener 1		5/25/2008 12:31:22	Y	N		
100035-01	0	5/25/2008 12:52:18	1	0	5/25/2008 12:52:18	400	999003	Complete IQ		5/25/2008 12:52:18	Y	N		
100041-00	0	5/25/2008 13:01:59	1	0	5/25/2008 13:01:59	200	999003	Complete Screener 1		5/25/2008 13:01:59	Y	N		
100041-01	0	5/25/2008 13:14:53	1	0	5/25/2008 13:14:53	400	999003	Complete IQ		5/25/2008 13:14:53	Y	N		
100039-00	0	5/25/2008 13:23:22	1	0	5/25/2008 13:23:22	200	999003	Complete Screener 1		5/25/2008 13:23:22	Y	N		
100039-01	0	5/25/2008 13:38:15	1	0	5/25/2008 13:38:15	400	999003	Complete IQ		5/25/2008 13:38:15	Y	N		
100038-00	0	5/25/2008 13:43:07	1	0	5/25/2008 13:43:07	200	999003	Complete Screener 1		5/25/2008 13:43:07	Y	N		
100038-01	0	5/25/2008 13:56:35	1	0	5/25/2008 13:56:35	400	999003	Complete IQ		5/25/2008 13:56:35	Y	N		

Appendix C. Frequently Asked Questions

Hardware/Software Requirements

Is the handheld charger and plug a universal standard or U.S. standard?

The default iPAQ power cord is based on U.S. standards. Each country will be delivered Universal AC Adapters, which will work in any country, for every handheld purchased.

Other than iPAQ handhelds, is additional hardware required for GATS data collection and consolidation?

Additional hardware requirements will be country-specific and dependent on the existing country IT infrastructure and the GATS survey operations and data management functions that need to be performed at the country level. The minimum hardware required at the national level for data collection and consolidation is:

- At least one Windows XP Service Pack 2 or later machine (standard laptop), 3 GB RAM, 250 GB hard disk.
- A backup device (such as USB memory keys or disks).

Multiple units meeting the above specification will be needed at each level of data aggregation if a given country aggregates data at intermediate locations in the country (for example at regional offices, or at the state level).

In addition to the GSS survey administration software on the iPAQs, are additional software requirements such as database software and software utilities required for GATS data collection and consolidation?

Software applications for data aggregation and some basic status reporting for GATS data collection and management operations will be supplied by RTI, GATS partner for IT services. There are no additional software requirements for countries transferring data via the iPAQ's SD cards. However, additional data transmission software will be required for countries utilizing fixed phone line or Internet-based data transmission. Countries transferring data via fixed phone lines will need a secure FTP (file transfer protocol) tool. RTI will select and test an open source version of this kind of product, which countries can then use. Countries transferring data via an Internet connection will need three additional pieces of software: (1) SQL Server 2005 Express at the National level for Web server support; (2) MS IIS for country Web site functionality and regional support; and (3) MS ASP.net for Web support. This software will not be supplied by RTI and countries should coordinate these software needs and any other software needs with CDCF, CDC, and WHO.

When will hardware and software support systems for deploying the iPAQs and aggregating data need to be in place and operational?

All hardware and software will need to be in place, tested and operational 30 days before the full implementation survey is undertaken. Testing of all hardware and software should begin as part of the pretest operations, and continue up to 1 month before the full survey.

iPAQ Administration Issues

What sort of computer knowledge and experience will interviewers need to successfully administer the survey on iPAQs?

The iPAQs and the associated GSS survey administration software is user friendly and requires that interviewers have only a minimum amount of computer knowledge to enter data and manage household assignments. More experience with computers is an advantage, but is not a necessity.

How durable are the iPAQ handhelds in rural areas where they may be exposed to a great deal of dust and where the survey may take place during the rainy season?

While some care should be taken to avoid exposing the iPAQs to water and dust, they are generally durable and have been tested for use in a variety of weather conditions. However, the iPAQs may fail if submerged in water. If there are concerns about the durability of the machines, there are many hard cases commercially available that can be purchased to provide dust proofing or water proofing.

Where have handheld computers been used before to do surveys and what are the types of difficulties countries have had in terms of the countries' capacity and capabilities?

Handheld computers have been used extensively in the United States and Europe and have been successfully used for surveys in Brazil and in several African countries. The availability of electric power is likely to be the most limiting characteristic for some GATS countries. The devices have a limited battery supply that should be recharged after every 8 hours of use. Countries will be issued additional batteries for the iPAQ to accommodate the needs of places with limited availability of power. Car chargers for the iPAQ are commercially available, but are not part of the countries' standard equipment issue.

What are the common problems one can face during the questionnaire administration in a field setting using this software?

In the field administration issues stem from two main problems—hardware failure and user errors. A small number of hardware failures are inevitable either because the iPAQ gets lost, dropped in a river, or some other unusual event. Therefore, countries will have spare iPAQs to deploy and should have procedures in place to send replacement iPAQs to interviewers in the field. User errors, on the other hand, are caused by interviewers' not properly understanding procedures or how to manipulate the handheld software. The GSS software was chosen for GATS precisely because of its ease of use and relatively low user error rate. In addition, all countries will receive detailed manuals explaining administration procedures and iPAQ use. Lastly, GATS partner organizations will collaborate with host countries to provide in-country training to ensure that interviewers and supervisors have the training they need to successfully administer GATS interviews.

What should be done to minimize the costs of iPAQ failing in the field?

Countries should plan to have a few additional iPAQs to use as spares. It is simple to load a new system with GSS software, personalize it for a given fieldworker, and to load cases to the machine. Any pending interview data from the failed iPAQ can be loaded to the new iPAQ via the removable SD memory cards. Each country should plan on the logistics of field replacements and estimate the time it would take to get a replacement to a field worker.

What are the advantages of the GSS software that will be used for GATS?

GSS offers a case management system and data transmission system. It is also strong at managing flow and skips within a questionnaire, thus allowing for a smooth administration. Furthermore, GSS is rather user friendly and requires that interviewers have only a minimum amount of computer knowledge to enter data and manage household assignments. GSS is easy to operate and has been modified to work in each country's native language.

What are the advantages of using handheld computers to administer the GATS survey versus paper-based administration?

There are three main advantages to administering the GATS survey using handheld computers that increase data quality instead of paper-and-pencil administration. First, the handheld computers' software has built-in validation checks to ensure that the interviewers enter valid responses only. If interviewers attempt to enter an out-of-range response, they will immediately get a prompt to choose among the valid answer choices. Second, the handheld computer manages the flow of the questionnaire and seamlessly routes the interviewer to the next question based on the respondent's prior answers. This enhances the flow of the administration of the survey and eliminates problems of skipped questions, questions asked out of order, and respondents erroneously receiving questions that are not logical based on prior answers. Third, data entry with handheld administration is immediate and simplified. Rather than interviewers writing down responses that data entry clerks later enter into a computer, interviewers directly enter answers into the handheld computer during the interview. Eliminating this extra layer of data entry greatly reduces the chances of data entry errors and allows for quicker delivery of the final dataset.

Data Transmission and Data Processing

What types of protocols have been developed for transmitting data from the iPAQ to the National Data Center? What sort of accommodations are there for countries in which frequent on-line data transmission may be costly or impractical?

RTI has outlined three basic data transmission protocols to fit a variety of needs and capabilities. RTI will collaborate with the country's IT staff to adapt these basic models to fit each country's specific needs. The three main data transmission protocols include: data transmission via the Web with a country housed server; data transmission via fixed phone lines; and transmission through the iPAQ's removable SD memory cards. In this third model, the SD cards can be copied or aggregated on a field supervisor's laptop and then electronically forwarded to a National Data Center or data can be physically sent to the National Data Center.

What sort of computer knowledge and experience will IT staff need to successfully load and deploy the iPAQs and manage the data aggregation process?

Prior to pretest implementation, RTI will train each countries' IT staff to use the hardware and software needed to support GATS data collection and aggregation process. At a minimum, IT staff at the National Data Center should have familiarity with running MS Windows based applications. In addition, some programming experience is recommended. Countries hosting a country Web site as part of their data transmission model will also need a small number of staff with programming skills in MS SQL Server, IIS, and ASP.net software, and staff with experience installing, configuring, and managing a public Web site. Countries aggregating data at regional centers (such as PSUs) will need additional staff at those

aggregation sites who are moderately skilled computer users with familiarity running Microsoft (MS) Windows-based applications.

How often will data from individual handhelds need to be transmitted and consolidated?

Several dimensions influence decisions about how frequently a country should upload data. They are:

- iPAQ capacity—RTI routinely uses an iPAQ for 100–200 cases with good results and have tested loads that simulate 350 cases for the GATS application (a database of 65,000 rows, a 10 MB database). Performance at the level of 350 cases was good with only some reduction in the speed at which the iPAQ performed some tasks.
- Risk management (i.e., protection against data loss)—although in theory, all interviews could be completed before aggregating data, we recommend that data be aggregated to the next level at least every seven days, regardless of iPAQ capacity. Uploading data daily would be ideal, but not necessarily practical. The iPAQ is a reliable device and loss of power does not cause data loss. Also automatic backups of the data are made to the SD memory card every time the field interviewer exits the case management system. But physical damage to the iPAQ or loss of the IPAQ can occur and result in the loss of data. Therefore, we recommend uploading data, which creates an off-site copy of the data, as often as is practical.

Survey monitoring and status reporting—the more frequent the transmission of data, the more up to date the survey monitoring reports based on the data will be. However, GATS protocol also includes guidance for a paper-based monitoring protocol to manage survey progress, which can be used in instances in which frequent data transmission is not possible.

Will data from the handheld machine require any processing for validation or consistency checks?

As data are collected, the GSS software performs validation at the instrument level per the questionnaire technical specifications. It should not be necessary to reapply the checks performed; however, review of the data is highly recommended. Data exported from the handheld will be ready for immediate review and subsequent processing as it is aggregated. Several GSS applications will be available for viewing the raw data and generating reports on the aggregated data. Additionally, tools will exist for writing the handheld (SDF) files to text/ASCII/CSV files so that country staff can perform further review and processing utilizing software such as SAS©, SPSS©, and STATA©.[1]

What kind of facility is there to make sure that data entered do not get corrupted and what are the data back-up specifications/configurations?

Upon completion of each interview, data are stored in two places on the iPAQ—on the iPAQs internal memory and on the removable SD card. These data are stored in an encrypted SQL database developed for iPAQ. This is a standard Microsoft product that has been well tested and well received. The iPAQ version of SQL is SQLce Compact Edition and it is a database format that works on a Pocket PC and is also usable on a standard desktop PC. Files can be moved from the iPAQ to a desktop or laptop and used by standard analysis software. A third backup of these data files is created each time the interviewer transmits data to the National Data Center.

[1] Use of trade names is for identification only and does not imply endorsement by the U.S. Department of Health and Human Services.

What sort of reporting tools will be available to track the performance and progress of the survey for individual interviewers, field teams, PSUs, and/or regions?

The exact nature of the performance progress reports will depend upon the frequency and method of the country's data transmissions. The database files generated by the aggregate of data from individual iPAQs are readable by a number of different analysis tools. It will be possible to develop computer-based reports with many standard analysis tools. Simple status reports will be available as part of the aggregation tools provided by RTI, but host countries may want to adapt these to develop their own custom reports. RTI has also outlined an example of a paper-based performance tracking report system for countries where data transmission may not be frequent enough to generate up-to-date electronic reports.

What are the primary differences between the GSS Suite and Enhanced GSS Suite?

The new Enhanced GSS Suite includes many features previously available to sites, but within an inclusive, easy to use project interface. The new GSS Suite includes menus for launching the new Questionnaire Designer, Database table editor, and case file tools. The system also provides a central location for launching the INI Tool, SDF builder, and data aggregation programs previously available to sites.

The structure of the databases used within the questionnaire has changed very little, and was unaffected by the development of the GSS Suite. The new GSS Suite gives users additional tools for viewing and updating questionnaires without using MS Access.

Creation of SDF files, previously completed within the INI tool can now be automated to run without the user interface. This is done by updating an INI file that the system uses to find the location of MDB and SDF files.

The data aggregation program includes modifications that allow XML files to be converted to SDF format. This feature is used in sites not using GSS to collect data.

What updates have been made in the July 2010 version?

In the July 2010 version the following updates or additions have been made:

- GSS Toolkit Additions
  - Reporting Utility: to provide tools for monitoring survey operations and data
  - Master File Merge Utility: To provide a tool for merging questionnaire data (IQ and HH), master sample file, and events data into one file for use by SAS and/or SPSS
  - File Builder program format updates
  - CMS Grid Designer: a utility added to allow design of the format and content of the columns of the CMS case grid

- GSS Engines additions
  - Added BackUpTo logic statement to allow backing up under program control
  - Updated FastForward
  - Updated Core Questionnaire programming for HQ and IQ
  - Updated Result Codes and auto coding rules for HQ and IQ

- CMS Updates
  - Added color coding for case status in grid
  - Updated format and content of Case Info and Case Details screens
  - Added data base utility to repair corrupted SDF files (GATS_DBUtil.exe)

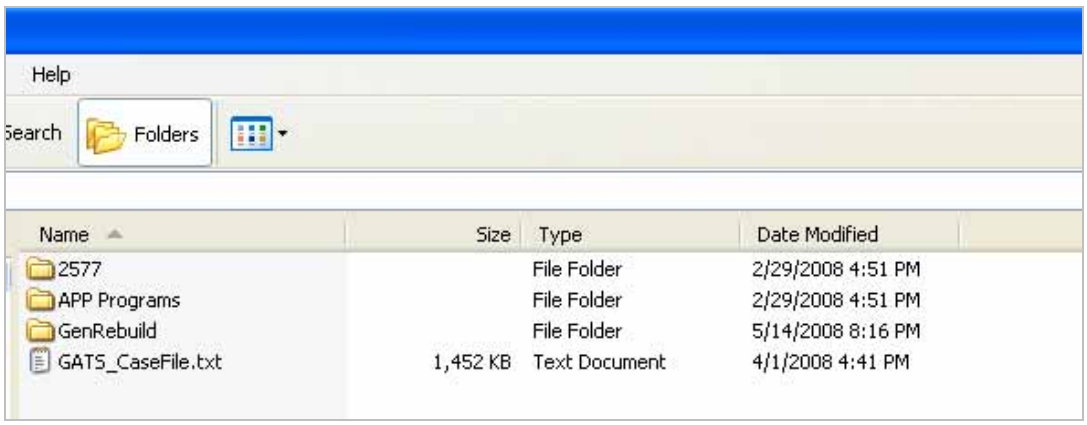
Task	Old Suite	New Suite
Questionnaire development	All sites used MS Access and worked directly with the raw database tables for questionnaire development. This Designer was modified to accept the updated database structures needed to handle more than two languages.	The new GSS Questionnaire Designer will make questionnaire development and modification much easier by allowing the user to update questionnaire text, logic, ranges, question types, and coded responses from a single easy-to-use screen. The Designer allows the user to add, modify, copy, and delete questions. Other features include the ability to sort questions, automated version control, and a new feature that allows the viewer to preview how the question will look on the handheld.
Data aggregation	This was completed using a stand-alone system that required the user to enter the location of the files to be merged with a master database.	The previous data aggregation system is incorporated into the new GSS Suite. It is called from a menu.
Transpose	Transposing data was completed using SAS, SPSS, or MS Access queries.	The new GSS Suite includes a utility that transposed data. In addition, the new system will produce a codebook and example programs in SAS or SPSS.
Case file	The casefile is a file containing household information required by the handheld to assign cases to Field Interviewers. The casefile was created using either SAS or MS Access. The casefile that is needed for the handhelds is a TAB delimited file with no text qualifiers produced in Unicode. This file is exported from MS Access or generated using a SAS program. In addition, it can be created using MS Excel.	The new GSS Suite has a utility for importing, editing, and generating a casefile.
Editing database files	The previous method required MS Access and updating raw database tables.	The new GSS Suite includes an interface for opening and editing database records. The new system allows users to edit tables that are needed for the GATS system. It prohibits updates to tables that are static or are not used for GATS.
SDF file generation	Previously, users were required to generate SDF files individually using the iniTool. This tool allows users to edit the INI table and generate an SDF file. It required the user to select the MDB to convert and to type in the project password.	The new GSS Suite has a utility that generates all SDF files required for the handheld. It is completed by selection a menu item. The files (MDB and SDF) are stored in an INI file that the user sets up when starting the study. The password is inserted automatically, and is not typed by the user each time the SDFs are generated.

Appendix D. List of Acronyms

AID	U.S. Agency for International Development
ASCII	American Standard Code for Information Interchange
ASP	active server pages
CDC	Centers for Disease Control and Prevention
CDCF	Centers for Disease Control Foundation
CE	Compact Edition
CMS	case management system
CMSDB	case management system database
CSV	comma separated values
DU	Dwelling Unit
DUEVT	Dwelling Unit Event Table
FTP	file transfer protocol
GATS	Global Adult Tobacco Survey
GB	gigabytes
GSS	General Survey System
GTSS	Global Tobacco Surveillance System
HP	Hewlett Packard
HTML	hypertext markup language
HTTPS	hypertext transfer protocol secure
iPAQ	Pocket PC personal digital assistant
MDB	Access format database file
NIST	National Institute of Standards and Technology
PC	personal computer
PDF	Portable Document Format
PSU	primary sampling unit
QID	Question ID
SAS	Statistical Analysis Software
SD	secure digital
SDF	SQL Compact Edition database files
SIP	Soft Input Panel
SPSS	Statistical Package for the Social Sciences
SQL	Structured Query Language
SQLCE	SQL Compact Edition
SSN	Social Security number
TCP/IP	Transmission Control Protocol/Internet Protocol
URL	uniform resource locator
USB	universal serial bus
VB	Visual Basic
WHO	World Health Organization
XMIT	Transmission Program
XML	extensible markup language
XP	Windows Operating System Version XP

Create Master Flash Card

Make the flash card by copying the three folders and one file to the root of an empty SD card. These folders and file should be on your PC's hard drive and include these folders and files.



To move files to an SD card on the iPAQ, unpackage an iPAQ, install the battery, and charge it.

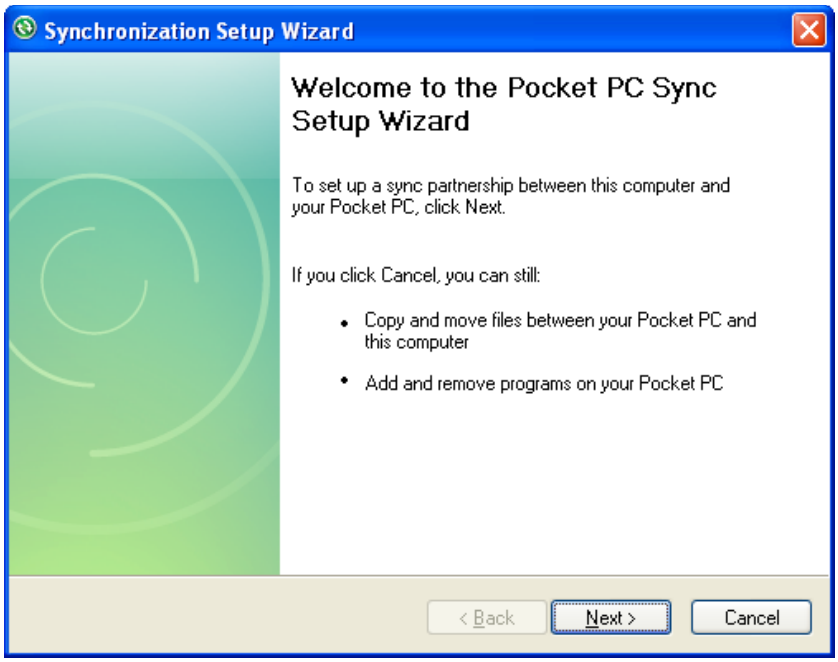
If the iPAQ is not new, run a factory reset to erase all data—make sure you see "Formatting PS…." Refer to the iPAQ documentation that comes with the device for instructions on how to complete a hard reset.

Get one of the SD cards and place it into the iPAQ.

Turn the iPAQ on and follow the setup instructions that appear on screen. Do **NOT** set the time or set a password on the handheld.

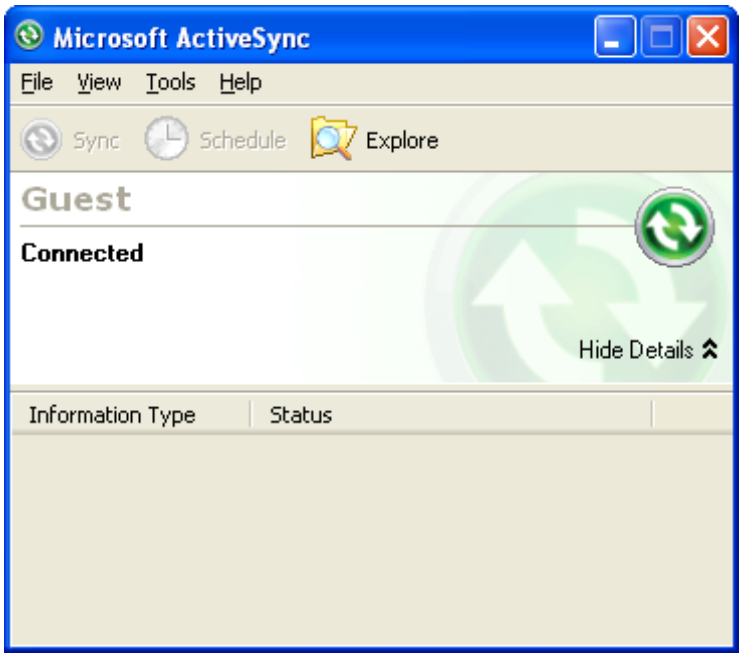
After installing ActiveSync on your PC, connect the iPAQ to your laptop using the USB cradle. This screen will appear.

Click on **Cancel**.

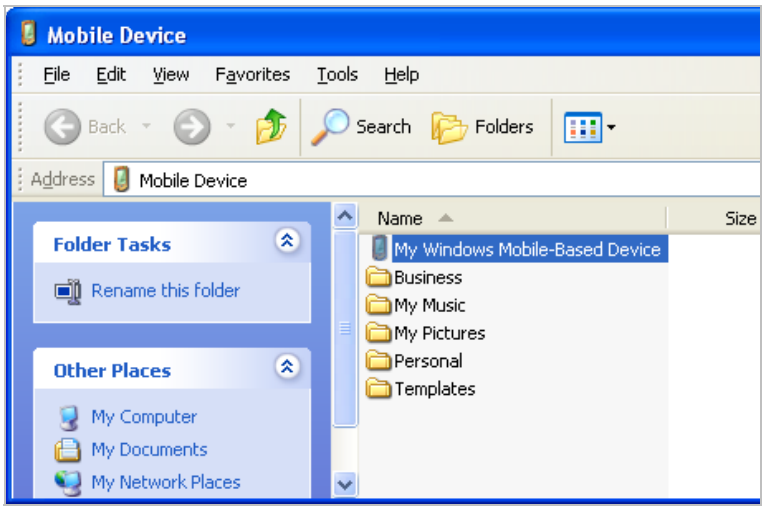


This screen will appear.

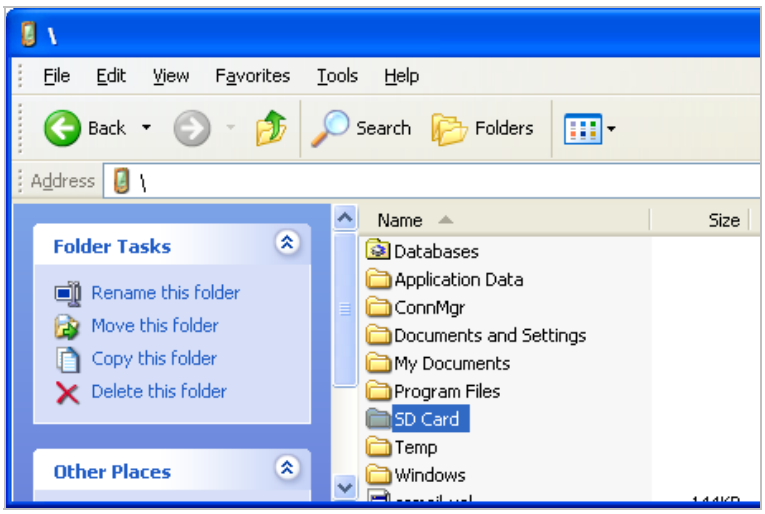
Click **Explore**.



Double-click on **My Windows Mobile-Based Device**.

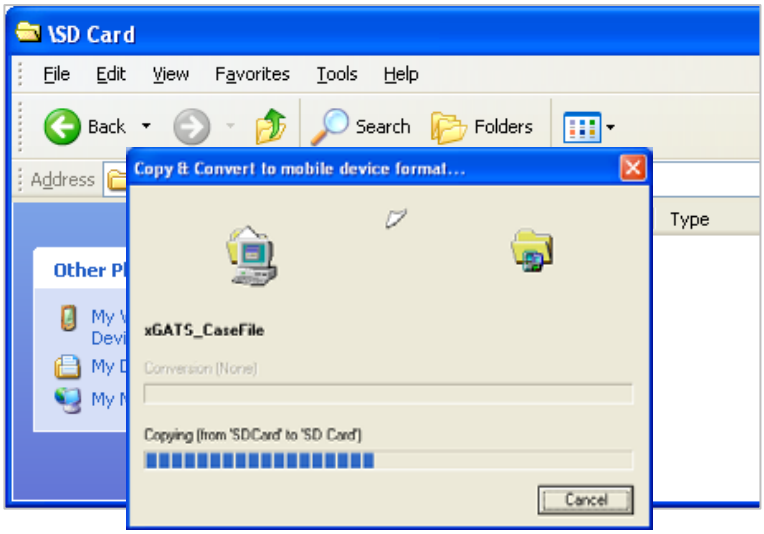


Double-click on **SD Card**.

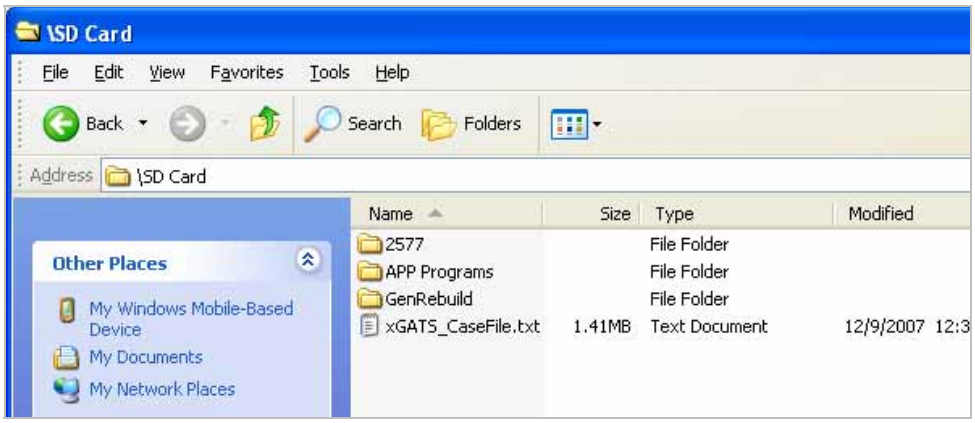


Drag and drop the files you unzipped onto the SD card.

You should see a window like this while the files copy.

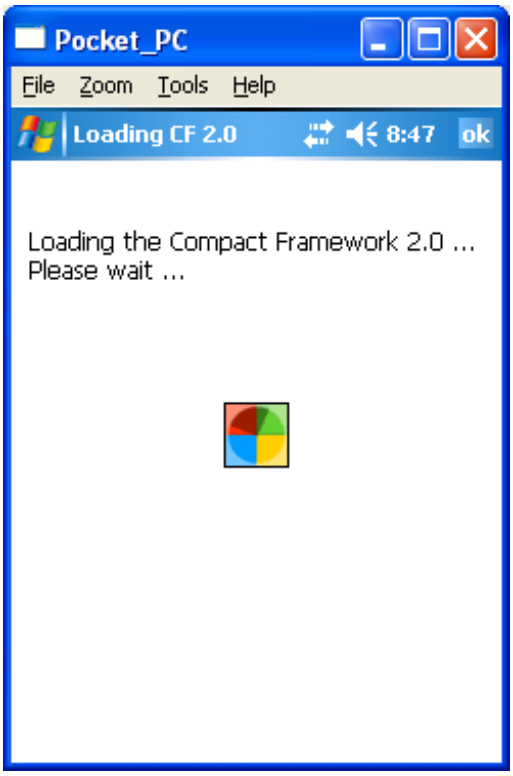


Your screen will show the contents of the SD card on the iPAQ.

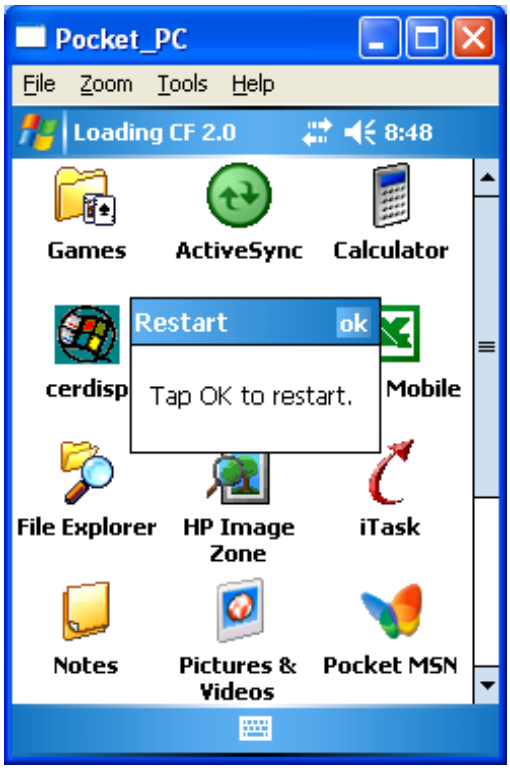


Turn the iPAQ off then on again. This will launch the "autorun" program to install the GATS software.

The first software to load will be the Compact Framework; this will take 45–90 seconds.



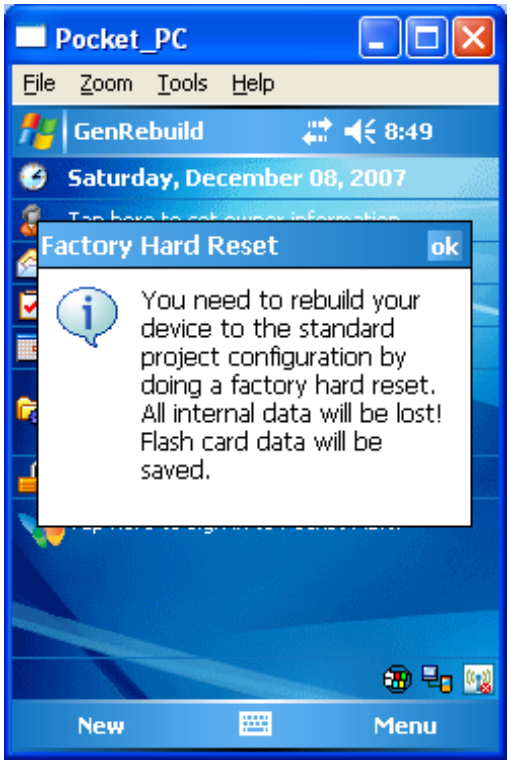
Tap **ok** after the Compact Framework loads.



The system will reset.

Tap **ok** to Factory Hard Reset message.

.

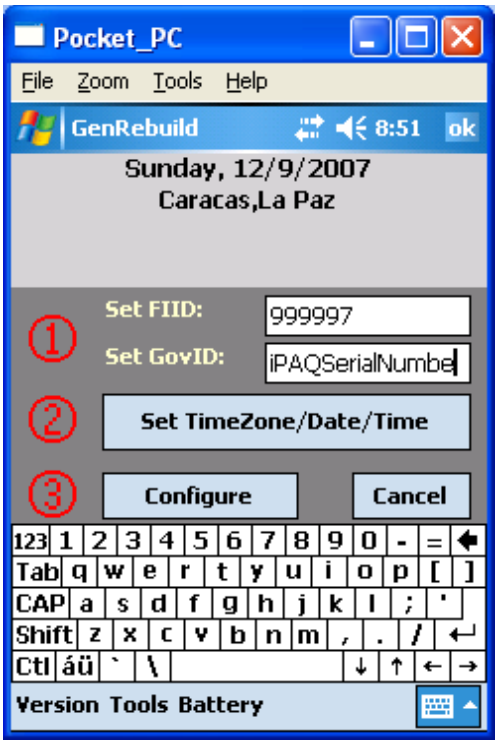


The Main Rebuild screen will appear.

Enter the following data:

- FI ID—Six digits; use 99997 for demonstration.
- IPAQ serial number or identifier.

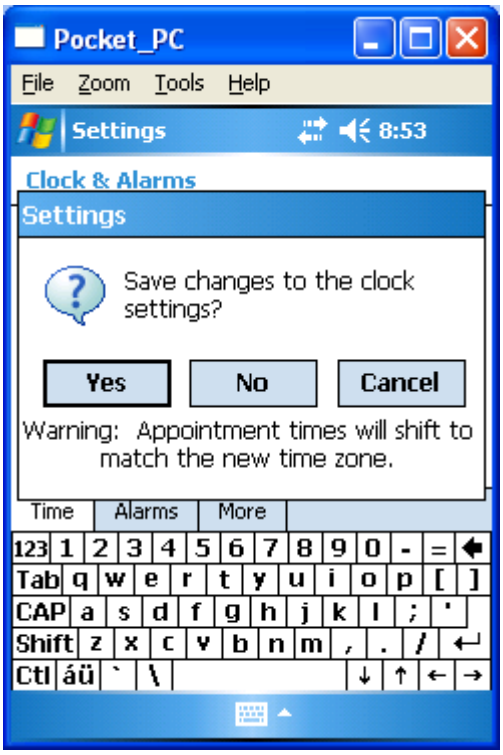
Tap **Set TimeZone/Date/Time.**



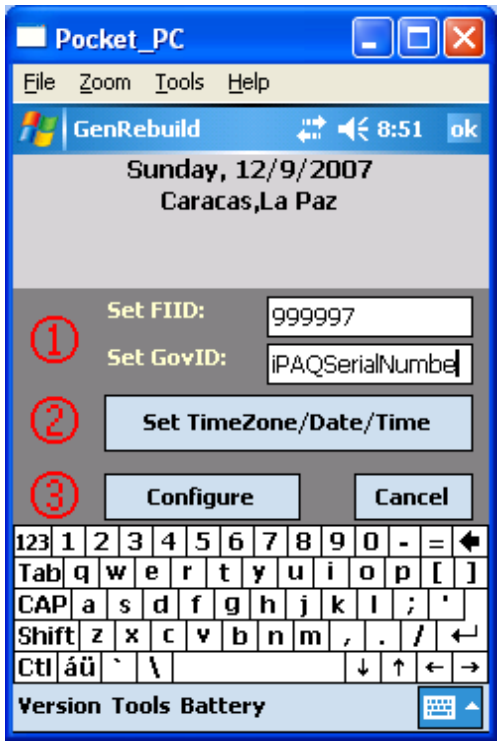
Enter the following data:

- Set time zone (critical).
- Set time and date (MM/DD/YYYY) and verify.

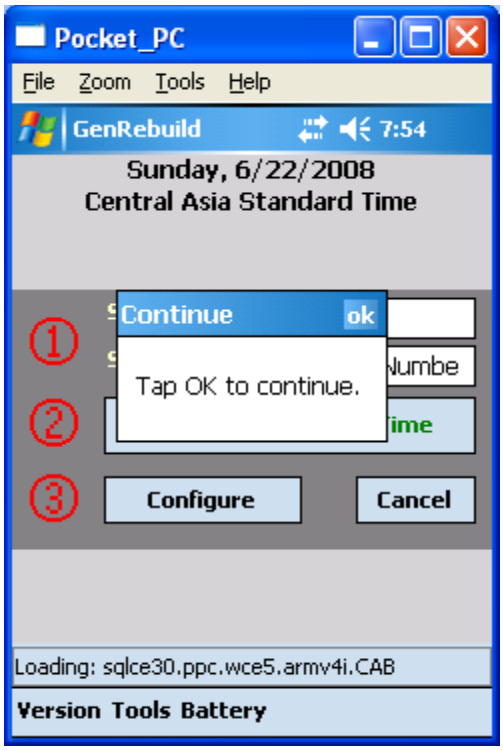
Save changes to the clock settings (Yes).



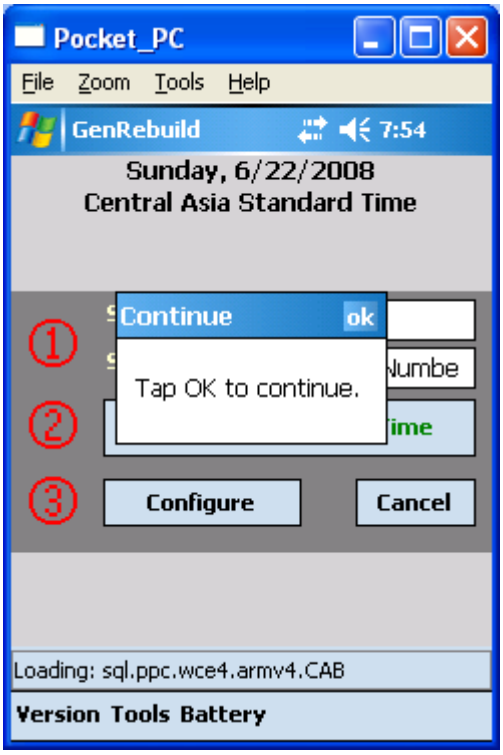
Tap **Configure**.



Tap **ok** to continue.



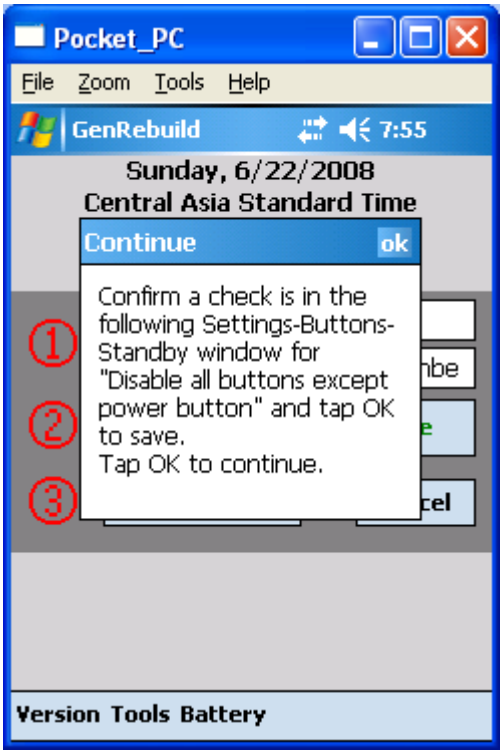
Tap **ok** again.



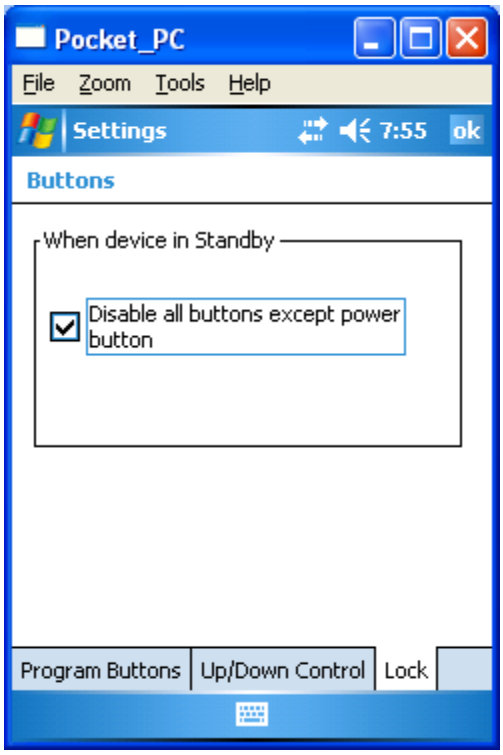
And tap **ok** again.

You will now set the device to disable all buttons except the power button.

Tap **ok** to continue.

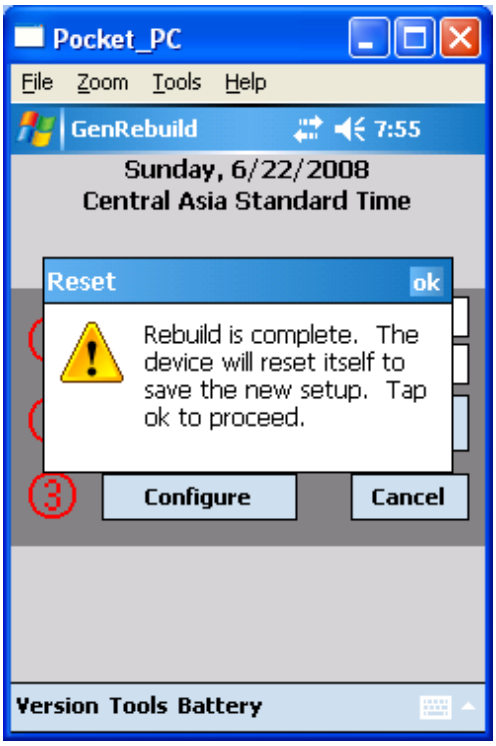


On the **Settings > Buttons** screen, make sure box next to "Disable all buttons except power button" is checked. If not, tap it to add the check mark and tap **ok**.



You will see **Rebuild is complete** message.

Tap **ok** and the system will reset one last time.



You are now prepared to run some of the following QA steps.

QA Steps

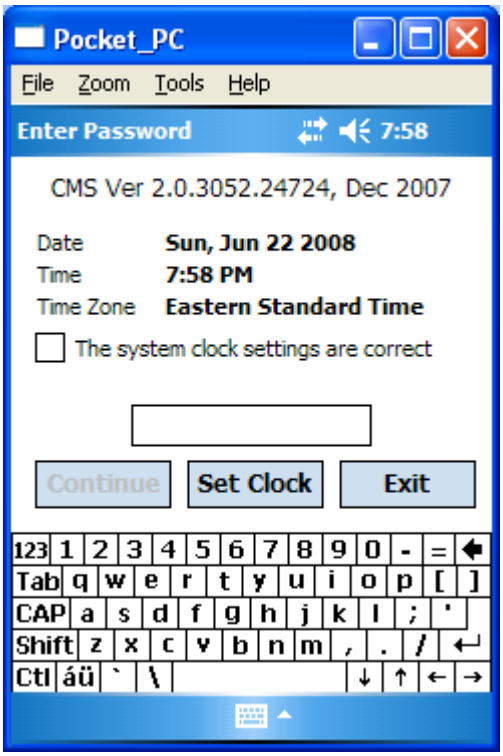
- Confirm date, time, and timezone.
- Confirm CMS starts (CMS password = gats); confirm CMS version (to be set before pretest).

Click on **Start**.

Click on **CMS**.

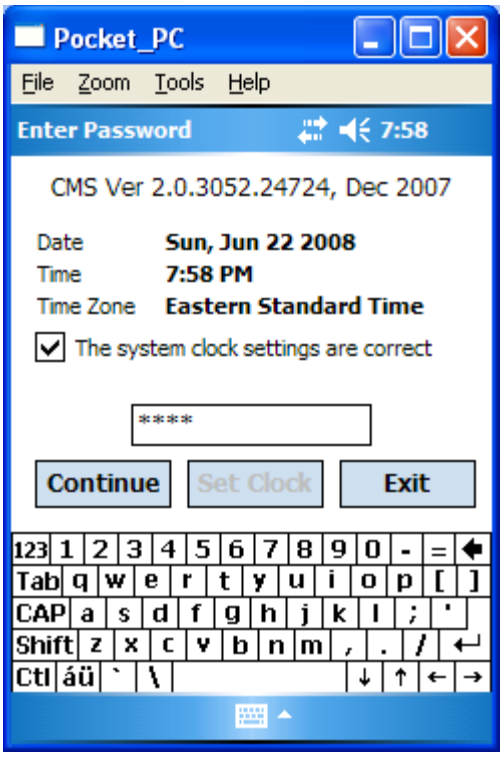
If date and time are correct, check the box next to "The system clock settings are correct."

.

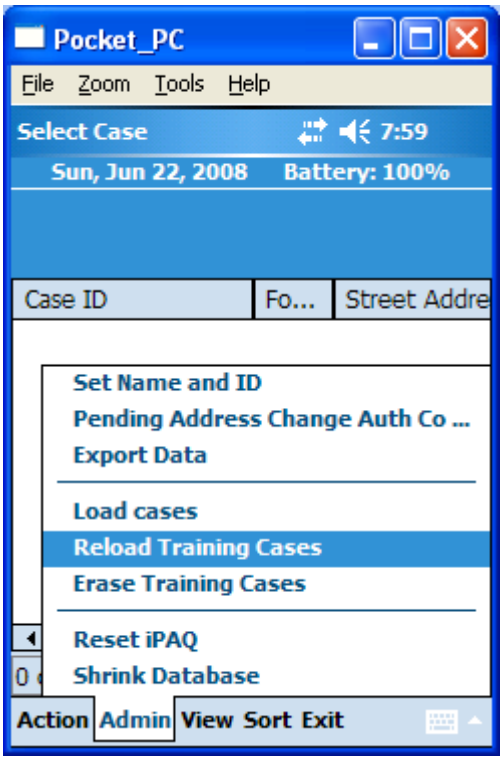


Enter the password “gats” for pilot study.

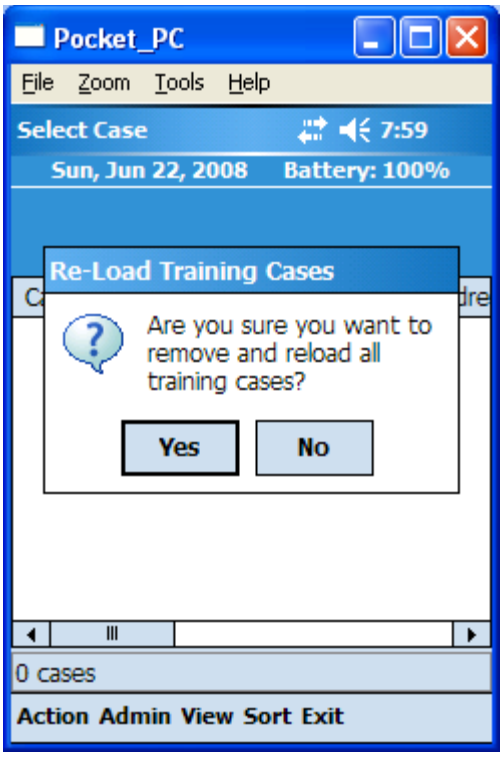
You will now load training cases for testing.



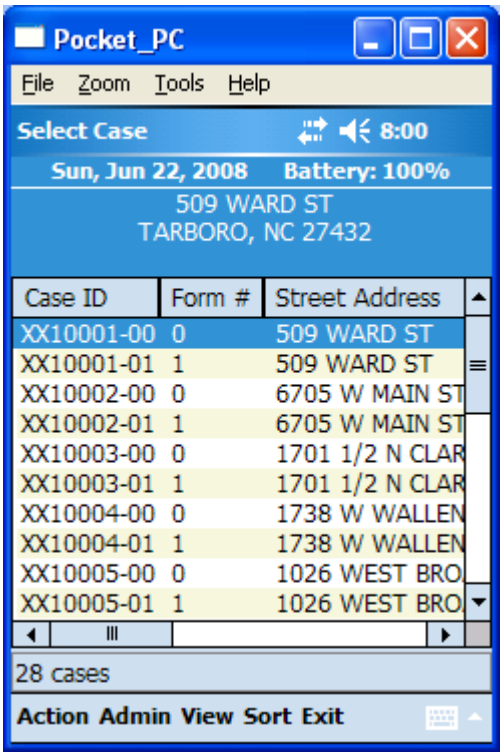
Click on **Admin → Reload Training Cases**.



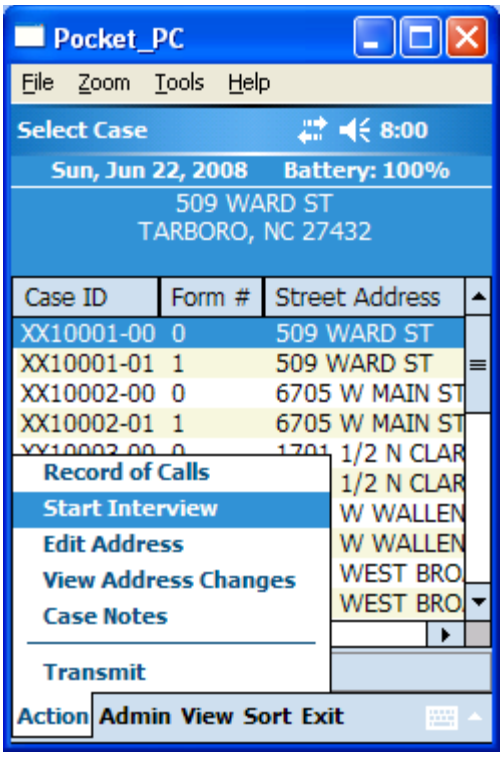
Click on **Yes** to load training cases.



Run GSS and open a questionnaire by clicking on **Action**.

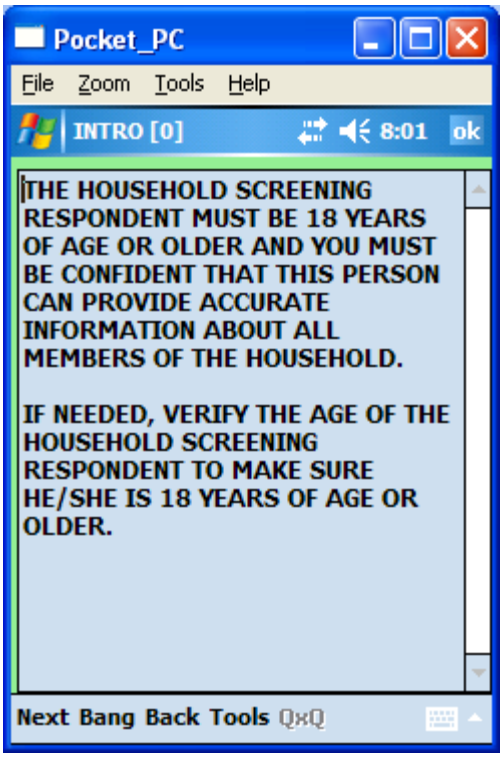


Click on **Start Interview**.



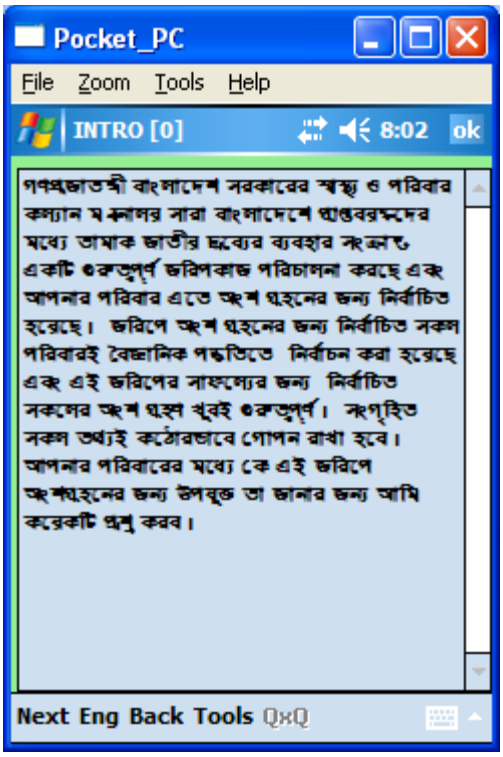
You will see the first screen of the questionnaire.

This screen shows an example of when a 2nd language is present. Click on **Bang** to see the current Bangla translation.

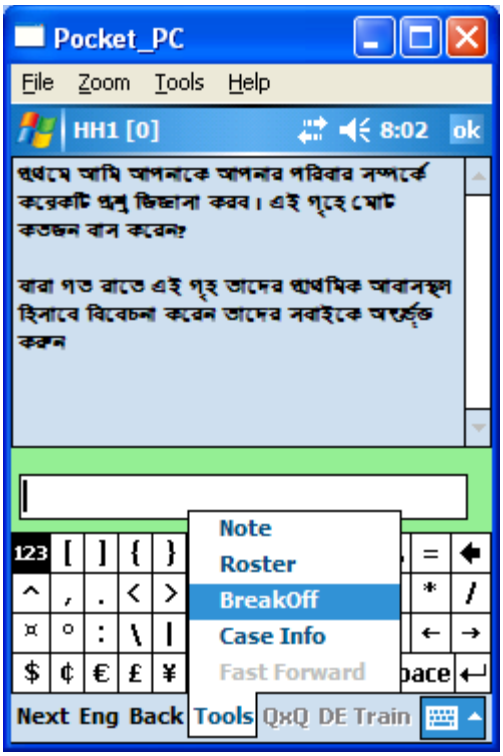


Now you will see the 2nd language text and font.

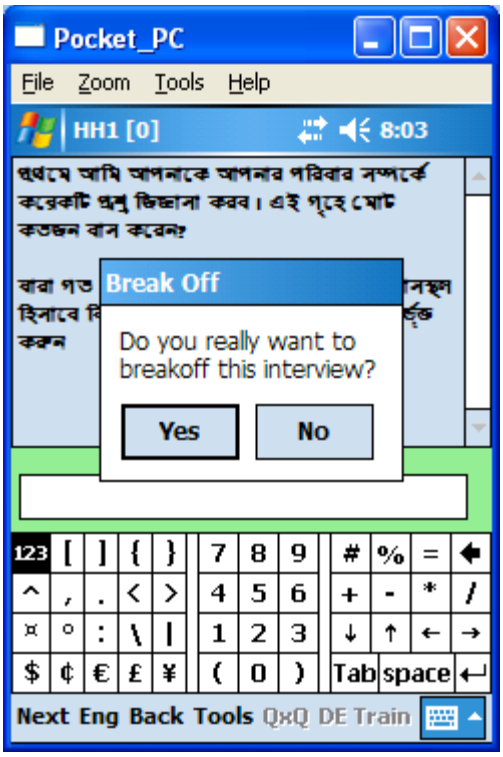
Click on **Next**.



You can break off from a questionnaire by clicking on **Tools → Breakoff**.

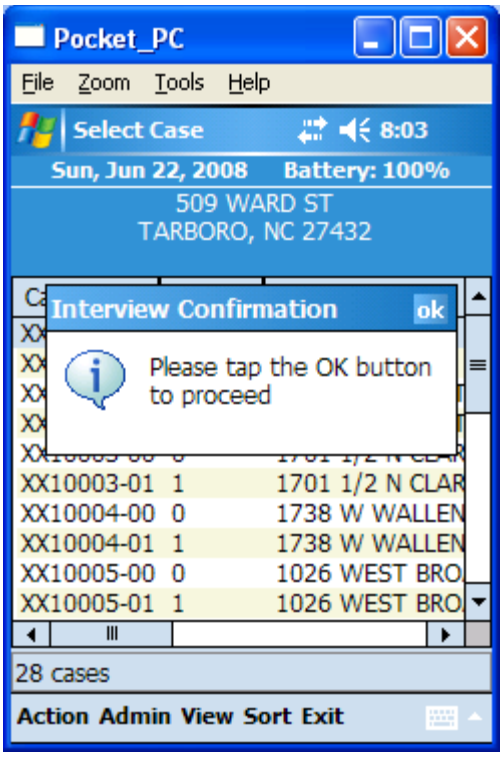


Click on **Yes** to breakoff.

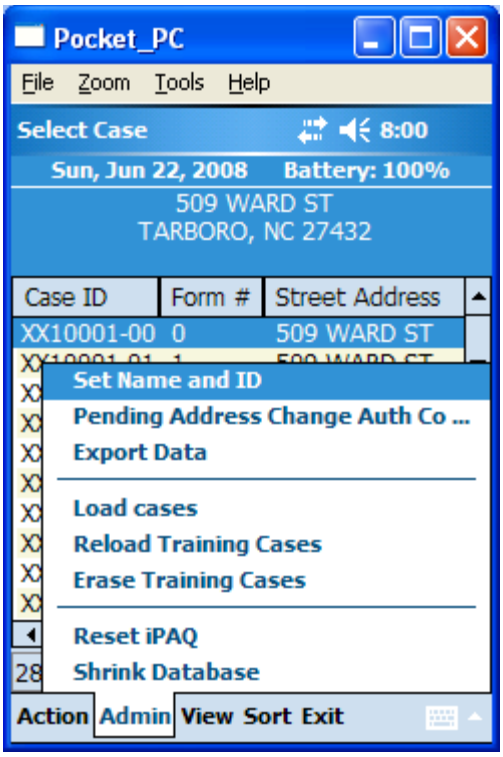


Click on **Exit**.

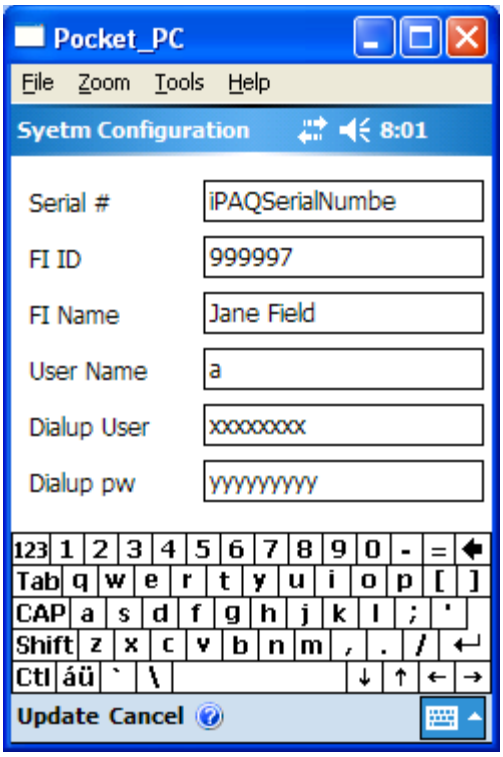
Click on **ok**.



Check setup parameters in the CMS (**Admin → Set Name and ID**).



Confirm setup parameters.



Appendix F. GSS Installation into an iPAQ Model 210

Start by completing a hard reset on the iPAQ model 210. The hard reset on the 210 is accomplished as follows:

Follow these steps to hard reset of the device. Press and hold the **Record**, **Windows**, **OK**, and **Reset** buttons at the same time.

Figure F-1. Buttons Used to Perform Hard Reset

1. **1 – Record**
   2 – Windows
   3 – OK
   4 – Reset
2. Hold all the buttons for 2 seconds.
3. Release the **Reset** button while continuing to hold the **Record**, **Windows**, and **OK** buttons.
4. A Clean Boot message is displayed in green text, indicating successful restoration of factory settings on your device.
5. Step through the start up procedures required after the hard reset to align the screen.
6. Slide the SD card with installation files into the iPAQ.

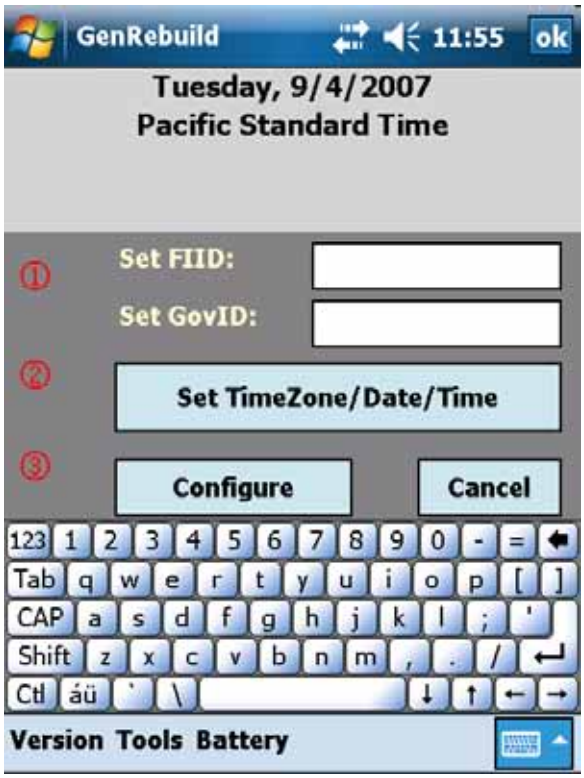
Enter **Yes**.

Enter **Yes**.

Enter **gats**.

Click on **ok**.

Enter **FIID** and **GovID** and click on **Set TimeZone**. Do not click on OK.



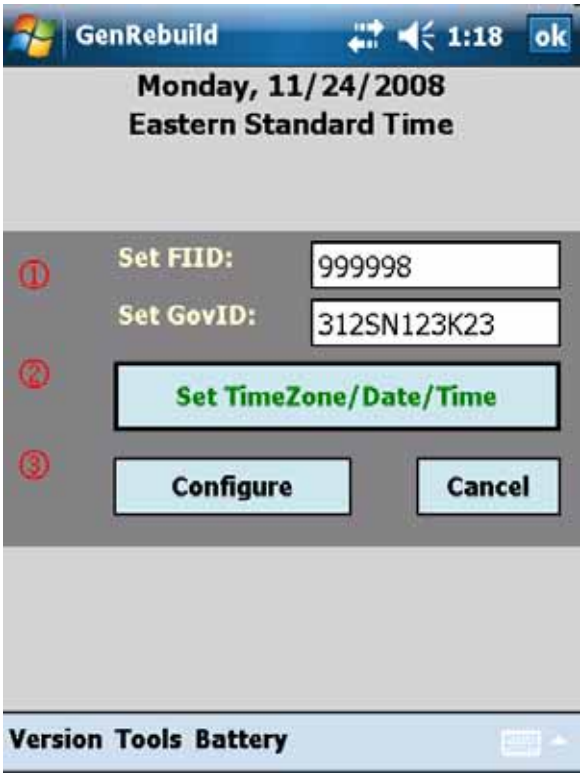
Click on **Set TimeZone**.

Select **time zone**.

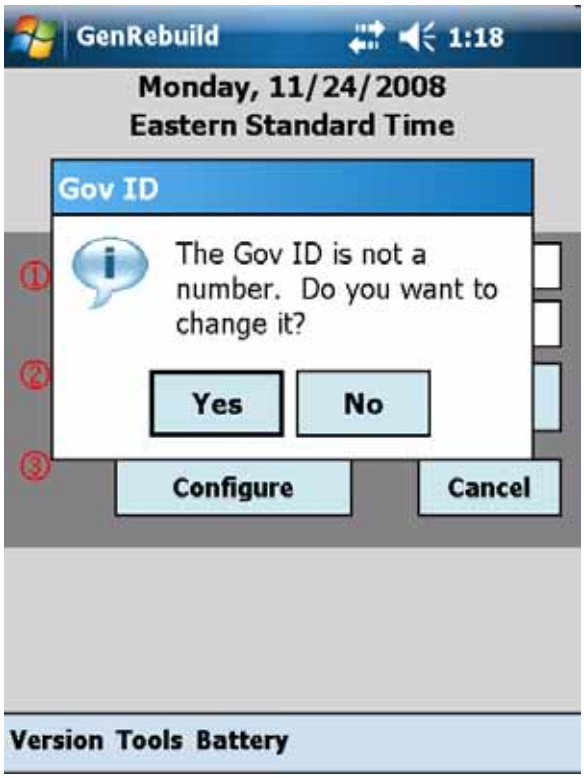
Enter Time and Date using AM\PM and MM\DD\YYYY format. Click on **ok**.

Click on **Yes** to save Date and Time settings.

Click on **Configure**.



Click on **No**.



Click on **ok**.

A soft reset will automatically be run on the handheld after this last load step.

The iPAQ will have this desktop.

To run the CMS, click on **Start → CMS**.

GLOBAL TOBACCO SURVEILLANCE SYSTEM (GTSS)