UNOHAC

United Nations Office for Humanitarian Assistance

SHAMAN

System for Humanitarian Assistance Management

System/User Manual

Preface

UNOHAC

The Office for Humanitarian Assistance Coordination (UNOHAC) in Mozambique was established as part of the UN operations in Mozambique (ONUMOZ) to increase the overall effectiveness of the emergency and peacekeeping operations. By incorporating a broad-based coordinating function, it was hoped that humanitarian assistance activities could be directed from a firm information base.

As a first objective, UNOHAC in Mozambique set about developing procedures and tools for information gathering, sharing, and dissemination.

The primary document resulting from this objective is the Consolidated Humanitarian Assistance Programme. This programme identifies sectors in need of assistance, i.e., agriculture, health, education. Needs within these sectors are detailed by activity, and each activity has a price tag attached to it.

In Mozambique, the programme document has been used to make a rational appeal for donor funds to match specific needs. As funds have been pledged or committed, UNOHAC has been able to provide donors and potential donors with up-to-date information on remaining needs, identify areas of over or under funding, and track the actual obligation of funds.

The methodology applied in the formation of the Consolidated Humanitarian Assistance Programme can be used at any stage of the continuum - from natural or man-made disaster, through rehabilitation and reconstruction to development.

An integrated database system, SHAMAN: System for Humanitarian Assistance Management, was developed to manage the mass of information in the Consolidated Programme, to keep it dynamically updated, and to produce reports in publishable form at any time. The database was expanded from the basic data, i.e., needs and contributions, to include other information useful for coordination of humanitarian assistance projects. For example, SHAMAN includes data on

all humanitarian assistance activities being implemented in the country by various agencies , population data, and incidents impacting humanitarian assistance activities

Because SHAMAN proved to be useful to UNOHAC in Mozambique, it was decided to turn SHAMAN from a Mozambique-centered database into a generalized database customizable for any country.

SHAMAN is a powerful and comprehensive integrated database management system It can be used as a planning and management tool, and as an implementing monitoring tool in connection with:

- disaster response and mitigation.
- rehabilitation in the wake of natural or man-made disasters.
- reconstruction planning.
- long-term development programming.

The principles of the CHAP can be applied anywhere - humanitarian assistance appeals, planning, operations, and development. SHAMAN is a convenient and tested tool for planning and managing humanitarian assistance and related activities.

Maputo, Mozambique
January, 1994

Table of

Contents

Part IV Appendices

List of

Illustrations

Quick Start

How to Install and Run SHAMAN

If you want instant gratification, don't like to read a manual or have only 5 minutes, these instructions are for you

1. Ensure that there is a line in the CONFIG.SYS file that reads:

```
FILES=171
```

and that the *largest executable program size* is at least 550K. This may be ascertained at the DOS prompt by typing \DOS\MEM. DOS version 6.x is recommended. If there is not enough memory, use DOS's MemMaker to optimize DOS's use of memory.

2. Re-boot the computer.

3. Insert the SHAMAN Program diskette in the A: Drive.

4. Type A: <ENTER>

5. Type INSTALL <ENTER>

SHAMAN will construct a number of sub-directories off the C: drive and decompress SHAMAN files into them. The program will then attempt to start and will present the user with the LogOn Screen asking for a user name and password. Accept the user name <SHAMAN> and type in the password <shaman>. SHAMAN will next construct index files for all SHAMAN tables and support files.

SHAMAN's initial installation contains complete data on Mozambique. It can be used to experiment with the program. When the main menubar is shown on screen, use the arrow keys to navigate from one choice to another. Press <ENTER> to select a choice. Repeated <ESC> will eventually exit SHAMAN.

To re-enter SHAMAN type: CD\CLIPPERS\ARTFUL\SHAMAN

About This Manual

This 4-part manual provides full information for understanding, using and maintaining SHAMAN.

The manual explains how to use SHAMAN to manage your humanitarian assistance resources. Every step, from inputting data to producing meaningful reports in varied formats is described. Information for managers, novice users, and experienced programmers is included.

The material is organized in four main parts:

1. SHAMAN Overview
2. Using SHAMAN
3. Technical Reference
4. Appendices

Part 1 is especially useful for managers wanting to assess how they can best use SHAMAN. It explains SHAMAN concepts, its database modules, its output in the form of reports or maps, how to customize SHAMAN to suit your situation, and technical requirements for using SHAMAN.

Part 2 provides the user with all information needed to access and use SHAMAN. Clear instructions and examples are given for every SHAMAN operation.

Part 3 is a technical reference for the system administrator. It includes instructions for system installation, system customization, and system maintenance.

Part 4, Appendices, includes all program code, data dictionary, authority tables, and sample reports, maps, and forms.

PART I

SHAMAN Overview

Introduction

About Part I

Part I of the SHAMAN manual provides an overview of what SHAMAN is and how it can be used to manage a humanitarian assistance program.

Managers can get a broad understanding of the SHAMAN system, to decide whether the database is suitable for their needs, to determine what equipment and staff they would need to use the system, and how they can customize SHAMAN.

MIS staff can use this section for an introduction to SHAMAN before they turn to the technical details.

Part I is divided into 4 chapters:

Chapter 1: About SHAMAN - What is the SHAMAN system?

Chapter 2: SHAMAN's Database - What database modules does SHAMAN have?

Chapter 3: Customizing SHAMAN - How can I customize SHAMAN for my situation?

Chapter 4: Technical Specifications - What equipment and staff do I need to use SHAMAN?

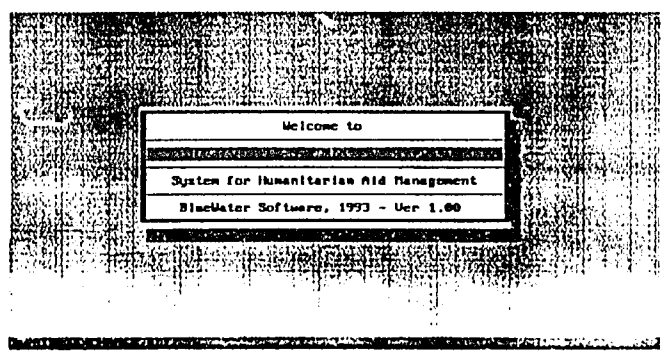


Figure I:1 - SHAMAN opening screen

Chapter 1

About SHAMAN

SHAMAN is a database system for humanitarian assistance management

SHAMAN: System for Humanitarian Assistance Management is a powerful database management tool. The database:

- relates financial data to humanitarian assistance projects.
- gives instant access to country-wide information on government and non-government humanitarian assistance activities.
- provides the ability to produce customized reports in the form of text or maps.

SHAMAN was developed for use in situations where humanitarian assistance is being provided by a variety of donors, governments and private organizations. A coordinating body can effectively use SHAMAN to maintain a dynamically updated information database on all aspects of humanitarian assistance and produce a wide variety of reports for dissemination to the donor community and all involved agencies. An organization could also use SHAMAN to focus on humanitarian assistance projects within the framework of its own operations.

SHAMAN can be used as a planning and management tool, but also as an implementing/monitoring tool at any stage of humanitarian assistance, from emergency to development.

SHAMAN Concepts

SHAMAN relates financial data to humanitarian assistance needs and current projects

SHAMAN was designed with two main factors in mind:

- the relationship of humanitarian assistance needs and activities to funding.
- management's need for easily accessible, up-to-the minute reports in varied formats.

The heart of the SHAMAN system is the Consolidated Humanitarian Assistance Programme.

This programme contains data on:

- humanitarian assistance needs,
- all funding, committed and obligated,
- designated sectoral activities,
- donors and implementers.

Extensive text on activity objectives, target populations, and background is contained within each sectoral activity record.

The Consolidated Humanitarian Assistance Programme is easily updated to provide management with up-to-the-minute information.

Other, interrelated modules provide important, supplementary information. Included are:

- information on the humanitarian assistance activities of various agencies -- Who's Doing What, Where?
- country resources, such as health and education facilities,
- population data, both current and historic, from various sources,
- incidents affecting humanitarian aid distribution.

Up-to-date reports and maps can be generated quickly and easily. With the addition of Geographic Information Systems software, SHAMAN data can be output visually in the form of maps and graphs

The strength of SHAMAN is in its report generating capabilities. A suite of commonly-needed reports is accessible within the system. Up-to-date reports can be produced in a few minutes. Further reports can be developed and incorporated as needed.

With the addition of mapping software, visual reports in the form of maps or graphs can also be produced from SHAMAN data. For management, this view of data can provide new insights.

SHAMAN Architecture

SHAMAN uses data-driven technology to provide a flexible, modular system

SHAMAN is composed of a series of interrelated modules that are easily accessed from a pull-down menu system.

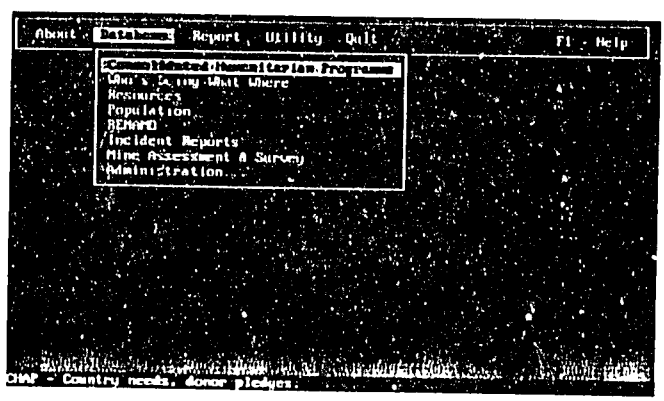


Figure I:2 - Main menu screen with pulldown database modules menu.

Design features include:

- Dictionary-based, data-driven technology.
- Open system architecture.
- Easily maintainable and modifiable modules.
- Powerful ad hoc query capability.
- Full-screen entry and editing of records.
- Extensive on-line help.
- Ability to generate new reports as needed or access standard reports from within the system.
- Ability to incorporate data with geographic information systems and output maps and graphs.

What you Need to Know

SHAMAN is a complex system with a simple and friendly user interface. Only basic computer skills are needed to use SHAMAN effectively

SHAMAN is a complex system with a simple and friendly user interface. For data entry, only basic keyboard and mouse skills are needed. Any reasonably-competent data entry clerk should be able to master the system with a few hours of use. Maintenance of the system requires a knowledgeable programmer, while modifications require an intimate knowledge of software used in SHAMAN system development.

SHAMAN Reports and Maps

A powerful report generator is incorporated in the SHAMAN system.

- R&R Report Writer® provides the ability to produce reports in varied text formats.

A built in suite of reports gives management immediate access to current information. Further reports can be designed and incorporated as needed.

- Reports can be output quickly in a standard format or in publishable quality.

The addition of MapInfo® (Geographic Information Systems software) allows SHAMAN data to be output visually in the form of maps and graphs.

- Information presented on country or regional maps allows analysis on levels not possible from textual data. See Part IV, Appendix E for examples.

Chapter 2

SHAMAN's Database

SHAMAN's database consists of a number of integrated modules, each easily accessible from the main SHAMAN menu

Each SHAMAN database module appears to the user to be a stand-alone segment, but in fact the parts are tightly integrated, sharing files (tables) with other modules. For example, the location table, shared by nearly all database modules, is the authority table for spellings in use for districts, towns, etc. This ensures that location names remain consistent across all modules, allowing valid comparisons, reporting, and mapping.

NOTE: Each database module contains a core set of files. The fact that some of these files are shared is transparent to the user, but integral to the design of SHAMAN. Any poorly-conceived system modification or maintenance can have a crippling ripple effect across the system.

Listed in this chapter are all standard SHAMAN database modules, including a general description, the data input and the reports output. Detailed information on the file structures and contents may be found in Part IV of this manual.

Consolidated Humanitarian Assistance Programme

The Consolidated Humanitarian Assistance Programme is the primary database module of SHAMAN. It allows tracking of donor contributions, matching them against sectoral needs to identify areas of over or under-commitment

The Consolidated Humanitarian Assistance Programme database module is the heart of the SHAMAN system. It contains records of all identified humanitarian assistance needs, donor pledges, and the $ needs attached to each activity. It matches donor contributions against sectoral needs to identify areas of over or under-commitment.

- Constantly updated, it allows an instant look at any activity to see current $ needs matched against total commitments to date.
- Memo fields allow full descriptions of activities and objectives.
- A powerful report generator outputs Humanitarian Programme information in publishable form.

A regularly published Consolidated Programme report is a powerful tool for soliciting donations, monitoring funding and activities, and responding to management, agency, and donor information needs.

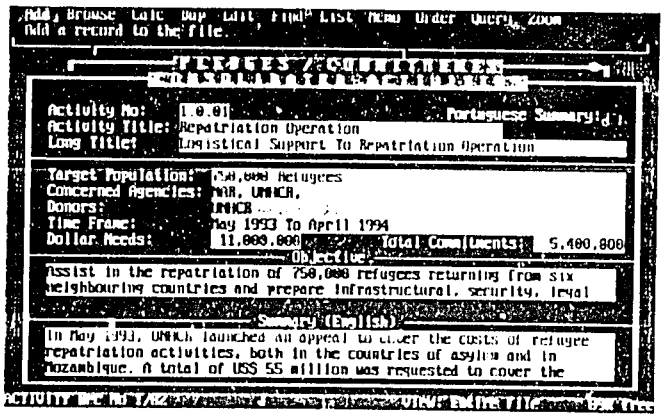


Figure I:3 - Consolidated Humanitarian Assistance Programme data-entry Screen

Data include:

- pledges and commitments from donors, government and non-government organizations, and implementing agencies.
- location, number and type of beneficiaries.
- $ needs and amounts committed.

Standard Reports include:

A regularly published Consolidated Programme report is a powerful tool for soliciting donations

- Consolidated Humanitarian Programme
- Contributions by Donor
- Contributions by Activity
- Contributions by Amount
- Contributions by Sector
- Financial Summary
- Donor Pledge Report Form
- Needs Summary
- Pledges/Activities

Who's Doing What Where?

The Who's Doing What Where Module provides information on the activities and responsible personnel of all of the humanitarian assistance agencies working in-country

The Who's Doing What Where? module documents the humanitarian assistance activities being implemented by agencies active in the country.

It lists all of the humanitarian assistance agencies working in-country and provides broad information on agency activities and projects. Included are all sub-offices, addresses, phone and FAX numbers, number of personnel and contact numbers, activities and their locations.

This module can be very useful to all concerned agencies for coordination of always limited resources.

Data include:

- agency names, addresses and contact names and numbers.
- agency activities.
- agency active districts.

Standard Reports include:

- Who-What-Where/Agency/Province/District
- Who-What-Where/ Province/ District/ Agency/Activity
- Who-What-Where Form (DeskJet)
- Who-What-Where Form (LaserJet)

Resources

The Resources module holds data on a wide variety of country sectoral resources, such as Health, Education, Water

The Resources module contains information on a country's resources in various sectors, e.g., health, education, water. Careful analysis of information needs before customizing this file can make it useful in a variety of situations. For example, it may be useful to know the number of schools in a district, the number of classrooms in each, and their exact location. This information can then be compared to population density, number of teachers, etc. and used as an indicator of the need for rebuilding or augmentation of sectoral needs.

Any type of quantifiable data concerning sectoral items can be input into the database. In the health sector, it can include information on numbers of hospitals or health centers, the number of beds per hospital, and the exact locations. In the water sector it might be useful to know the number of hand-dug wells.

Reports from data in this module are useful planning documents. Addition of mapping software can make this a powerful analysis tool.

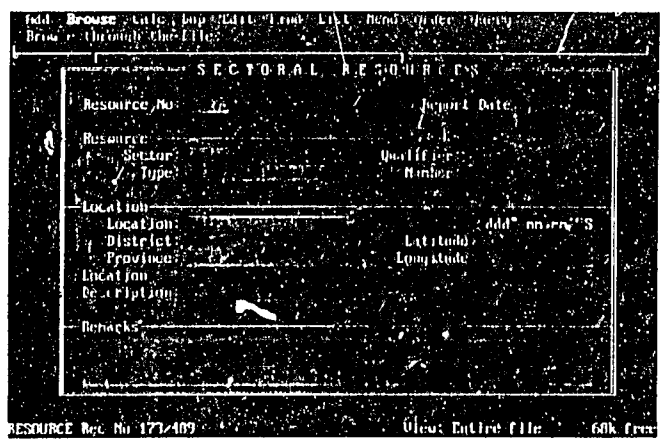


Figure I:4 - Resources module data-entry screen.

Data include:

- information about resource sector - e.g. education, health, water;
- resource type, number, and location.

Output:

- Customized reports and thematic maps.

Population

Population figures from varied sources can be compared by district or country in the Population module

The Population module holds population figures by district from various sources. It includes figures on the number of persons currently receiving humanitarian assistance benefits.

The source field names for population figures are customized at setup.

There are fields for up to five population estimates from government or non-government agencies and two fields for the latest government census figures.

Comparison of population estimates from varied sources can be done at a glance. Overall population can be summed up quickly and output to reports.

Information provided by this module is especially valuable for thematic mapping in the resource sector.

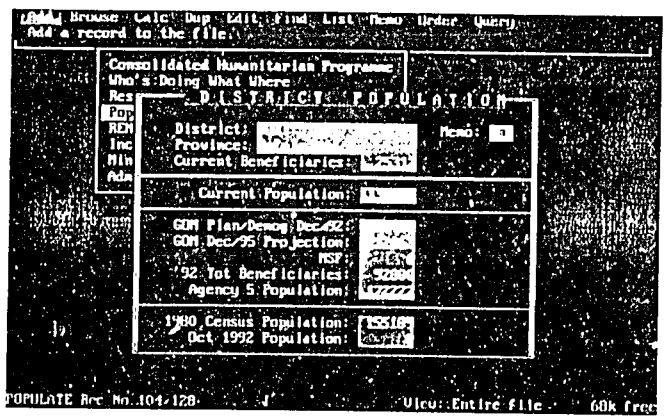


Figure I:5 - Population module data-entry screen.

Data include:

- Actual and projected population figures by districts

Output:

- Customized reports and thematic maps.

Partisan Dispatches

Dispatches of food and non-food items to disrupted areas can be monitored using the Partisan module

Partisan is used here in the sense of any opposition group, for example, RENAMO in Mozambique, the SPLA in Sudan, or UNITA in Angola. The file/module name can be customized as appropriate.

One result of disruptive conditions, such as civil war, is an information-gap on resources and conditions in disrupted areas. An agency or coordinating body can monitor food and non-food dispatches to these areas using this module.

Dispatches may be tracked or output by agency, by province, by district, by date, by commodity, and by amount.

Data include:

- dispatches of food and non-food items to disrupted areas by various agencies.

Standard Reports include:

- PARTISAN by Province District
- PARTISAN Dispatches by Agency/District/Category.
- PARTISAN Dispatches by Month/District/Category

Incident Reports

Reports of incidents impacting humanitarian assistance, such as accident, theft, or land mines, can be tracked and related to identify problem areas.

The Incident Report module provides a means to track incidents affecting humanitarian assistance activities, e.g theft of food aid, vehicle accidents, harassment, or land mines, and relate clusters of incidents to specific locations or activities.

With the addition of mapping software this module becomes a powerful analytical tool to determine any incident patterns or problem locations.

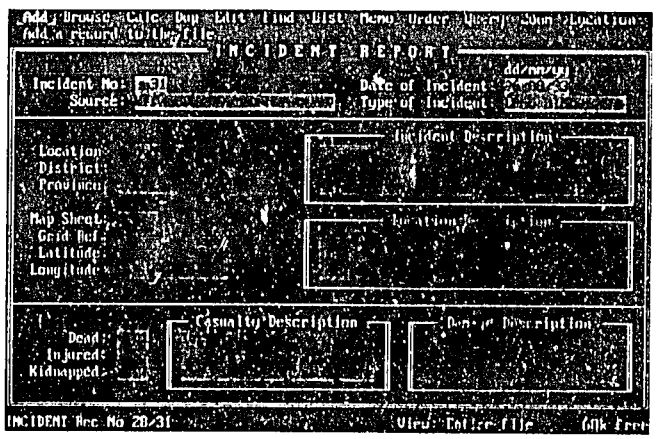


Figure I:6 - Incident Report module data-entry screen. Note the need for text editing procedures.

Data include:

- Incidents impacting humanitarian assistance activities .

Standard Reports include:

- Incident Report

Mine Assessment and Survey

Information on mined roads or other suspected mine locations, and the progress of de-mining operations, can facilitate safe delivery of humanitarian assistance and reintegration of populations scattered by warfare

The Mine Assessment and Survey module provides a means to track mine locations as they are reported, found, and removed. It can be used by de-mining teams and is also a valuable source of information for humanitarian assistance agencies working in areas mined or suspected to be mined. With the addition of mapping software, known dangerous areas can be quickly identified.

The 2-phase mine assessment and survey operation involves:

1. an initial qualitative assessment of danger areas with teams reporting on types, probable numbers and suspected locations of mines.

2. a detailed survey of actual mines and types found, along with records of their disposal.

This module details and interrelates both phases.

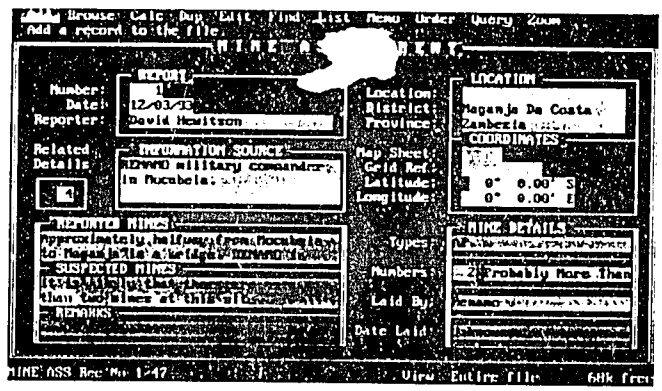


Figure I:7 - Mine Assessment and Survey module data-entry screen.

Data include:

- Mines reported and their locations.
- Mines found and their disposal

Standard Reports include:

- Mine Assessment & Survey

Administration

Dissemination of information is made easy by maintenance of a customized mailing list capable of producing mailing labels specific to a particular document.

The present version of SHAMAN contains one administration module consisting of a mailing list that allows up to eight document types to be entered. SHAMAN prints out labels for the specified mailing, mak-

ing dissemination of notices, monthly bulletins, or other publications quick and easy

The administration module can be modified to add files useful to an organization, e.g., a personnel file.

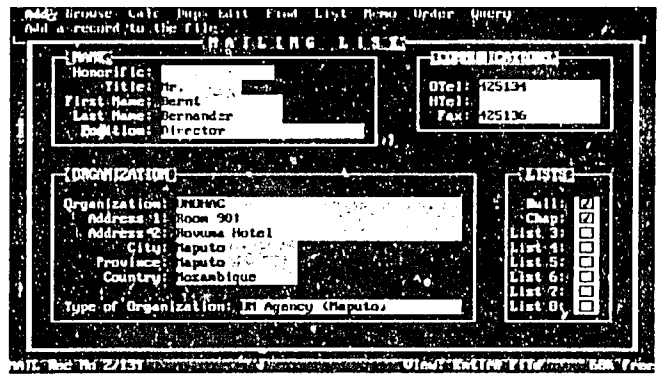


Figure I:7 - Administration Mailing List module data-entry screen.

Data include:

- Names, addresses, honorifics, phone and FAX numbers, type of organization, and what mailing list(s) each individual or group is on (up to eight categories).

Standard Reports include:

- Mailing Labels - Bulletin

Chapter 3

Customizing SHAMAN

SHAMAN can be customized to meet your humanitarian assistance management needs.

SHAMAN can be customized to suit the humanitarian assistance program management needs in various countries and of various agencies. SHAMAN was developed for a coordinating agency, UNOHAC, and will be most adaptable to the needs of such a body. However, the structure is such that it can be set up for other development, monitoring and reporting needs.

For successful customization, management must analyze its program to determine the basic data categories and information needs. A suggested approach is outlined below.

Some organizations might choose to use only part of the SHAMAN system. For example, it is quite possible to use only the Consolidated Humanitarian Assistance portion of the database. A mine assessment and survey team might want to use only the Mine Assessment module and customize only the related main and support files.

In any case, basic analysis must be done, preferably in cooperation with a competent computer database specialist, to ensure that needs mesh with technical constraints.

Analysis Phase

1. Preparing the Consolidated Humanitarian Assistance Programme

Careful analysis of the total humanitarian assistance needs is a crucial first step.

The categories, sectors, and activities encompassed by the Programme must be determined. In SHAMAN's 3-tiered system, one category will have multiple sectors; sectors will have multiple activities.

Activity Table Example

In the following example, the category 3 (Emergency Relief), has 4 related sectors, and 9 related activities. One sector (Relief Food Aid) with one of its activities (Nutritional Rehabilitation) is shown.

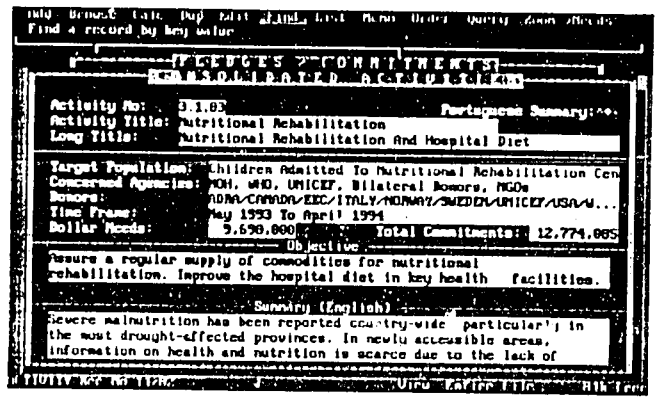


Figure I:8 - Activity File data screen for Activity Number 3.1.03

3	represents the Category	- Emergency Relief
.1	represents the Sector	- Relief Food Aid
.03	represents the Activity	- Nutritional Rehabilitation

The UNOHAC Consolidated Programme Activity file can be used as a model

Careful analysis of the Consolidated Programme before setting up SHAMAN is crucial for the smooth operation of the database. Any changes in the basic Programme will have a ripple effect across the system. However, accompanying text can be edited at any time as the Programme matures. New activities can be incorporated as well. Deleting former activities is problematic and should be avoided if at all possible.

The 3-tiered Programme information will be customized as SHAMAN's Activity Title File.

Attach $ amounts to each activity need.

The Consolidated Programme Activity file used by UNOHAC in Mozambique is reproduced in Part IV, Appendix C of this manual. It is a good model on which to base your Consolidated Humanitarian Assistance Programme.

2. Determine Needs

The second task is to attach $ needs to the activities identified in Phase 1. Each 4-digit Activity will have one or more needs records. For example, the 3.1.03 has two needs records:

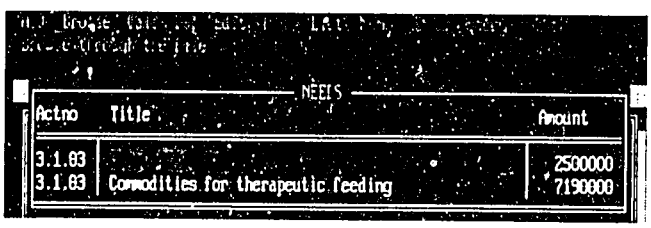


Figure I:9 - Browse screen for Activity Number 3.1.03 Needs records.

Changing the $ amounts of the needs later is a simple editing task.

3. Determine Resource Sectors and Types

Decide which resources you want to track

The Resource file has an associated support file (RES_TYPE) that controls the resource sector and resource type information. Analysis must be done on these two items before entering any resource data.

For example, assume that sector resources under Health, Education, and Water are to be tracked.

Under the sector heading of Health you may want to track health posts and hospitals. Two records will be entred into the Resource Type file as follows:

Sector	Type
Health	HEALTH POST CENTER
Health	HOSPITAL

Perform this analysis for all sectors to be tracked.

4. Determine Population Data Sources

Choose up to 7 sources for population data

The Population module allows comparison of population data from various sources by district. You must decide on categories/time periods useful to your organization, and the sources of the data.

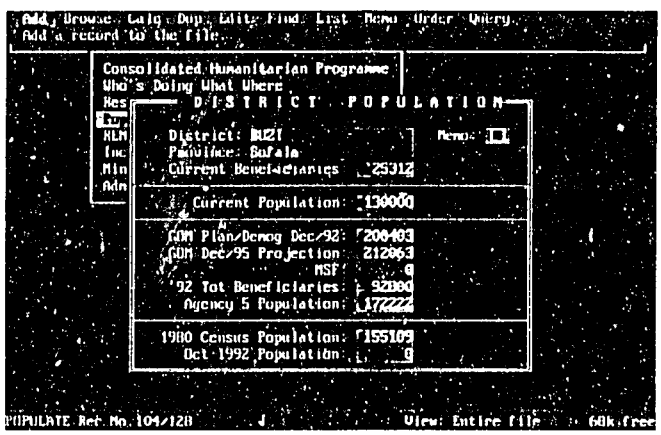


Figure I:10 - Population File data screen for one district showing bottom 7 modifiable categories of population data collection.

For example, you may wish to track MSF-generated data on district population figures. A natural choice, of course, would be the country's own census figures and projected population figures by district.

5. Determine Publications for Mailing List

SHAMAN allows up to eight publications to be included in the Mailing List module.

Choose publications that are distributed regularly to any group of recipients and enter them into the list.

For example:

Consolidated Humanitarian Assistance Programme
Bulletin
Monthly Report
Incident Report

6. Determine Report Needs

Determining report needs is an ongoing activity

This is an ongoing area for analysis. Standard reports are included in SHAMAN modules. However, management may have requirements for other reports from time to time. These reporting needs should be communicated to the system administrator. Virtually any report from SHAMAN data can be produced using R&R Report Writer.

The remainder of the customization phase is technical. Given the information base determined above, the SHAMAN system administrator can proceed with customizing the database fields, authority tables, etc. There will also be other routine customization requirements, such as country-specific field names.

Customization Phase

General instructions and suggestions can be found in Part III, Chapter 2, Customizing SHAMAN. Directions for using the Utility modules needed are in Part II, Chapter 4, SHAMAN Utility.

Chapter 4

Technical Specifications

SHAMAN is a responsive, modifiable and easily maintainable system.

SHAMAN is a turnkey system that uses hierarchical pulldown menus, extensive online help screens for the operators and ad hoc query capabilities. The programs are written in CLIPPER (a dBASE compiler) and ARTFUL II (a third-party database-engine) and delivered in the form of executable files. The basic system is operable by anyone with a few hours training.

Reports are initially prepared by using Concentric Systems' R&R Report Writer® for Clipper. Memory swapping is seamlessly performed by OVERLAY() or Blinker®.

SHAMAN is a *data-driven* program, i.e., much of its behavior is modified at runtime and is not predetermined by hard-coded data. Most of these data are kept in support files that are user modifiable while other data are in a data dictionary file which is itself modifiable at runtime. This makes for a more responsive, modifiable and easily maintainable system than is possible otherwise.

Operating System Requirements

Hardware:

SHAMAN is a mission-critical application. For best results, it requires powerful hardware.

For efficient operation, the system should *minimally* be installed on an IBM-compatible 33mhz, 80386 computer with a hard disk greater than 100mb and 2mb or more of RAM. However, it is recommended that a 33mhz, 80486 DX machine with hard disk greater than 200 mb and 8mb of ram be used as a server in a peer to peer Local Area Network. Lantastic® version 5 has been successfully used in this regard.

Peripherals:

Two other pieces of hardware are necessary to ensure the continued running of the computer: a voltage stabilizer and an Uninterruptable Power Source (UPS). The voltage stabilizer should have electronic stabilization for small fluctuations in voltage plus a servo-motor control to handle large fluctuations.

The STAC ST-500 has been successfully tested. It is placed in front of a UPS, i.e., it is connected directly to the mains. The ST-500 not only stabilizes the voltage, it also converts from 220 to 110 or vice versa and has output plugs for both voltages.

A UPS should be chosen with a capacity of running at least 500 watts for 15 to 30 minutes.

Computer Support Requirements

A capable data administrator will have a good knowledge of software programs used by SHAMAN and a basic knowledge of hardware maintenance.

SHAMAN is a *mission-critical* application designed specifically for humanitarian assistance program management. It was designed and tested in the field. SHAMAN is said to be mission-critical because it directly influences the effectiveness of program operations. As such, the system needs continuous availability, operational reliability, data integrity and security.

The above can only be supplied by a computer-literate data administrator with a good working knowledge of Clipper, DOS, and the other software packages that are part of SHAMAN. The person also needs a basic knowledge of computer hardware functioning and maintenance. He/she should be capable, for example, of swapping a faulty floppy disk drive or removing a non-functioning hard disk to ship out for repairs. The Data Administrator should direct all data entry and devise and implement procedures for data verification. He/She should be responsible for report generation and be able to design new reports on demand from management.

SHAMAN data entry can be handled by any reasonably competent data-entry clerk with a working knowledge of English.

Many validation routines are incorporated into SHAMAN. For example, in many fields, SHAMAN will not allow entry of data unless validated by authority files. However, SHAMAN data includes extensive text files and other qualitative data that depends on the integrity of data input. It is important that the data administrator understand the need for rigorous validation of data entry.

SHAMAN data entry can be handled by any reasonably competent data-entry clerk with a working knowledge of English. If SHAMAN data is enhanced with the addition of mapping software, it would be useful to have someone on staff with some familiarity with GIS systems.

Information Support Requirements

The effectiveness of SHAMAN as a tool for data manipulation and dissemination depends heavily on the methods used for data collection and verification. SHAMAN data will most likely be gathered from many agencies and by several agents. Standardization of data collection forms or formats is recommended as part of the customization phase.

The effectiveness of SHAMAN depends on the methods used for data collection and verification.

The forms have to be a compromise between what is easy for data collection and what is easy for data entry. For ease and accuracy of data entry there should be a relationship between the data collection form and the data entry screen. Sample forms can be found in Appendix F.

SHAMAN's associated reporting and mapping capabilities are powerful. It's worth taking extra time in analyzing data collection methods and forms to take full advantage of these capabilities.

PART II

Using SHAMAN

Introduction

About Using SHAMAN

SHAMAN: System for Humanitarian Assistance Management, is a robust and comprehensive database management system designed to meet complex humanitarian assistance coordination needs. It was initially designed for UNOHAC: United Nations Office for Humanitarian Assistance Coordination in Mozambique, but can be customized for particular countries and situations.

Although easy to use, SHAMAN is an extremely sophisticated and comprehensive database program. Because all the work is done "behind the scenes", the non-specialist user has a powerful tool at his/her disposal without having to know all the technical details.

SHAMAN is designed for the non-computer expert. A new user can learn to maneuver through SHAMAN and become proficient in entering data and producing reports in a few sessions. There is little or no danger of corrupting files or crashing the system through beginner errors.

A log-on system protects SHAMAN data by limiting access. Three levels of access are permitted: Novice, Expert, and System Administrator.

- A Novice user can look at the data but cannot change it.
- An Expert user can enter and edit data in most fields. He/She is locked out of the Data Dictionary and System File and any other files designated by the System Administrator.
- The System Administrator has access to all files and can modify any part of the SHAMAN system.

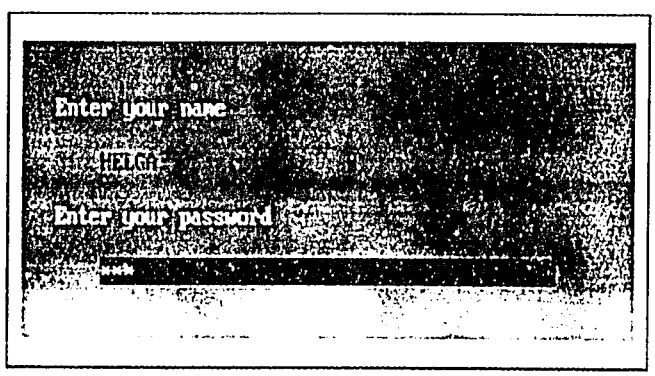


Figure II:1 - SHAMAN LogOn screen.

SHAMAN is composed of many interrelated modules. Some comprise several related tables for data entry; others serve as support units. To the general user, these interrelationships are transparent. The support units contain files which the system administrator uses initially when customizing SHAMAN, and subsequently for SHAMAN maintenance.

Part II of the SHAMAN manual is written for the user and provides full instructions for using Shaman's pulldown menus and doing full-screen maintenance.

Chapter 1: SHAMAN Menus and Maintenance Tools This chapter gives information on how to access and navigate SHAMAN menus and explains specific terms and skills needed for full-screen maintenance of SHAMAN files.

Chapter 2: SHAMAN Database Use this chapter to get information on SHAMAN database modules and for step-by step instructions for entering data.

Chapter 3: SHAMAN Report Use this chapter to get instructions for producing standard, integrated SHAMAN reports.

Chapter 4: SHAMAN Utility This chapter gives detailed explanations on the use and maintenance of SHAMAN System, Support, Utility, and External files and functions.

Chapter 5: Generating Reports Use this chapter for general instructions and needed references for using R&R Report Writer® to generate SHAMAN reports.

Chapter 6: Generating Maps Use this chapter for general instructions and needed references for using MapInfo® to generate SHAMAN maps.

Chapter 1

SHAMAN Menus and Maintenance Tools

This chapter gives instructions for navigating SHAMAN menus, and explains the skills needed to do full-screen maintenance.

It is divided into five parts:

Useful Terms
Navigating SHAMAN Menus
SHAMAN Action Menu
Using SHAMAN Action Tools
Using On-Line Help

Useful Terms

Access To call up a menu, an action, a screen, a record, a picklist, etc., by pressing the appropriate key(s).

Browse To scroll through multiple records to view or select for fur ther action.

Checkbox A box in which a √ can be toggled on and off.

Child Record A record that is dependent on a parent record, e.g., in the Who What Where module, a What record is dependent on its parent Who record.

Default A set value which appears automatically in a field. It can be overwritten.

Dialogue Box A message that pops up on screen asking you to per form some action, e.g., "Press any key to continue."

Dropfield A short list of the acceptable values (names or numbers) which can be called up from a field. Values are usually hard-coded in. Dropfields in SHAMAN are identified by a down arrow beside the field.

Field A single unit of information in a database record, e.g., an Activity Number.

Full-screen Maintenance The ability to do maintenance on records by having them appear one record per screen.

Light Bar A movable bar or block of light used to highlight a menu item or field.

Maintenance Performing any action on a record, e.g., add, edit, duplicate, etc.

Menu Bar The top of the SHAMAN screen containing menu choices listed horizontally.

Module An organized grouping of related data.

Parent Record A record that has dependent children, e.g., a Who record has a related What record. This is its child record.

Picklist A menu (list) of acceptable values (names or numbers) which pops up or can be called up from a field to select a choice. Usually from an authority table and may have more than one column.

Pulldown Menu A secondary menu that appears onscreen when you access a menu.

Popup Box A box containing information or values that pops up on screen, either automatically or through direct access.

Screen Prompt A dialogue box or message that pops up on screen to give you information.

Scroll Move the Light Bar horizontally or vertically through the menu bar or the SHAMAN screen using the arrow keys.

Select Scroll to highlight your choice and press <ENTER>.

Support File Often a file used as an authority table for spellings, values, etc.

Table A DBF (database) file of records (rows) containing loosely related fields (columns).

Utility A peripheral routine that helps you perform some task, e.g., backup files.

Value A name or number entered into a field.

Zoom Move from the record in view to a related record of another table.

Navigating SHAMAN Menus

All SHAMAN records and actions are accessed through pulldown menus. The opening SHAMAN menu gives the user the following choices:

About Database Report Utility Quit F1=Help

About: Shows SHAMAN Credits

Database: Gives access to all SHAMAN database modules

Report: Gives access to: SHAMAN Reports...
Printer Support
Prepare Memos for Reports
Update Activity File for Reports

Utility: Gives access to: System ...
Support Files ...
Utilities ...
External ...

Quit: Closes all files and quits SHAMAN

F1 = Help

To select a menu item:

- use the left or right arrow keys to highlight an option and press <ENTER>

OR: Enter the first letter of your choice, e.g., <D> for Database.

A second level of options appears in a pulldown menu:

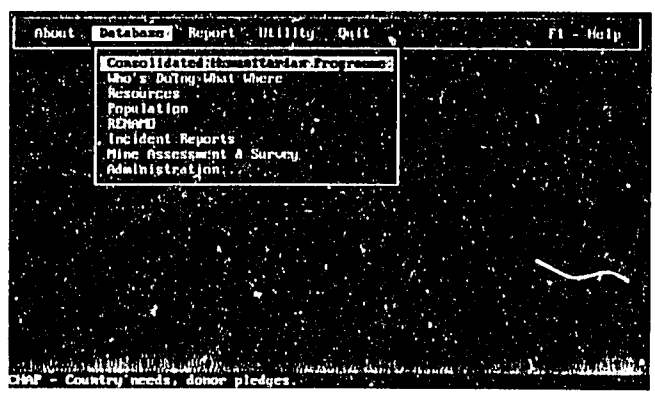


Figure II:2 - Database pulldown menu.

The extension "..." indicates a second level of pulldown menu choice:

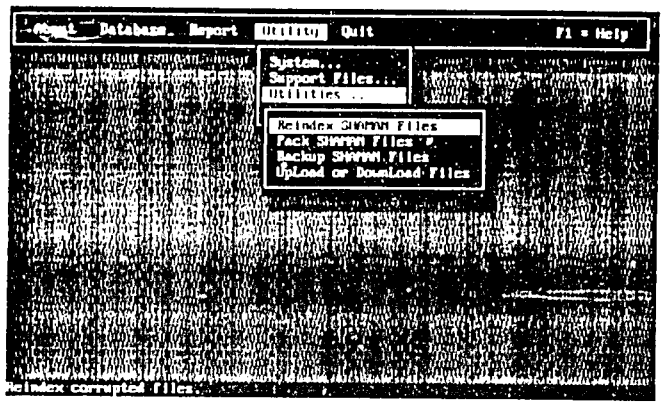


Figure II:3 - Utility menu showing a second pulldown level.

To select a pulldown menu item:

- use the up or down arrow keys to highlight an option and press <ENTER>.

The menu bar across the top of the screen changes to give a choice of actions.

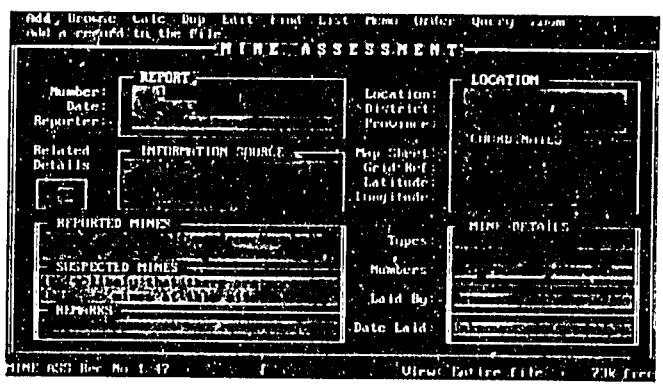


Figure II:4 - Mine Assessment screen showing top action menu bar.

Action Menu

SHAMAN's action menu (full-screen maintenance menu) gives the user the following choices:

Add Browse Calc Dup Edit Find List Memo Order Query Zoom	
Add	Add a new record
Browse	Browse records in the table in use.
Calc	Count: Count the number of records in the table in use. Sum: Sums up all numeric fields in the table in use.
Dup	Add a new record starting with a duplicate of the record in view.
Edit	Edit all Fields: Edit all fields in the record in view. Edit Non_Key Fields: Edit non-key fields specific to the record in view.
Find	Search for specific records in the table in use or in a related table.
List	Select a report related to the table in use.
Memo	Add, view or edit notes on the current record.
Order	Select the order in which to view the table in use.
Query	Query the table in use or a related table.
Zoom	Move to a related table.

The action menu bar will include only the choices applicable to the table in view. For example, Zoom will appear in the menu bar only if it is possible to move to a related table.

To select a menu item:

- use the left or right arrow keys to highlight an option and press <ENTER>

OR: Enter the first letter of your choice, e.g., <A> for Add.

Using SHAMAN Action Tools

Add

Choosing Add brings up a new, empty record of the table in use, with the cursor on the first field. Entry of data is straightforward, with the following safeguards and assists in place:

- Usually, a new record comes up with *new* inserted in the first field. SHAMAN fills in this field with a prime identifier when the record is saved.
- Date convention: Dates must be entered in the British convention of dd/mm/yy,
  e.g., 27/09/94
- Picklists: SHAMAN provides picklists for fields where data must match information in authority tables. This safeguard ensures that standard spellings, acronyms, etc. are used. If an unacceptable value is typed in, a picklist pops up. Scroll the list to select the correct value and press <ENTER>. The correct value will be entered in the field.

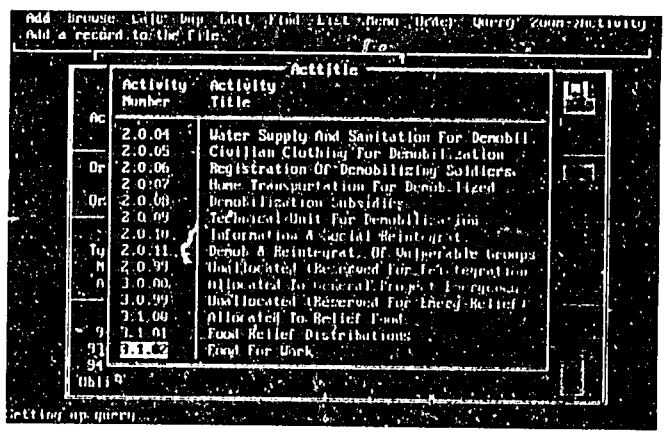


Figure II:5 - Activity No. picklist from Activity Title file.

Alternatively, the user can go directly to the picklist by pressing <ENTER> on the empty field. This option makes data entry quick and easy.

To call up a picklist on a field with data already in it, type in any letter and press <ENTER>. Because the value is now unacceptable, the picklist will pop up.

Fields provided with picklists can be identified onscreen. As the field is highlighted, an outline box is drawn around it. Normally, picklisted fields cannot be left empty. In a few exceptions, a screen prompt will appear instead of an outline box. Type in any letter and press <ENTER> to access the picklist. Otherwise, press

<ENTER> to move to the next field.

- Dropfields: A few screens contain dropfields. (Dropfields differ from picklists in that the choices are hard coded and cannot be changed in the Support Utility). Dropfields are identified by a down arrow next to the field. To access the choices in a dropfield, press <SPACEBAR>.

- Checkboxes: Some screens contain checkboxes. Use <SPACE-BAR> to toggle the √ on and off.

- Checkboxes are also used to show that a memo exists for a record. The √ is toggled on by SHAMAN when a memo is entered.

- Saving records: SHAMAN uses a screen prompt to remind the user to save a record.

  OR: Save the record with a <Ctrl W> command.

- Deleting records: If you've made an error, or don't wish to save an entry, press <ESC> to go back to the previous record. To delete a record once it has been saved, press <DELETE>. A screen prompt will pop up for you to confirm your choice.

Browse

The *Browse* option is available for a quick overview of records. In *Browse* mode the user can view multiple records but cannot change them.

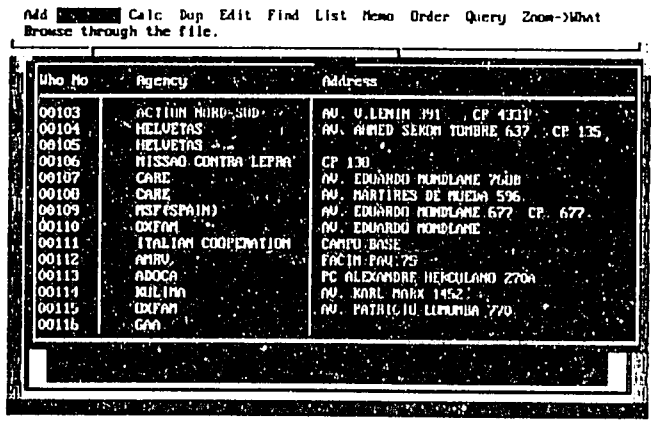


Figure II: 6 - Browse screen from the Who module, showing Agency addresses.

- To view records, scroll vertically or horizontally.

- To access a record for full screen maintenance, scroll to highlight the record and press <ENTER>.

- To change the order of the records being viewed, select *Order* before you *Browse*.

- To return to the original record screen press <ESC>.

Calc

The *Calc* function is especially useful when used in conjunction with a sub-set of data selected with *Query*.

Calc: Count quickly counts the number of records in the sub-set of records selected.

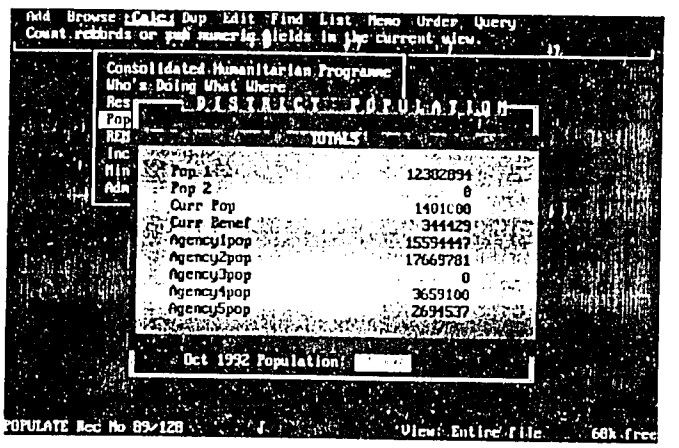


Figure II:7 - Calc/Sum screen showing total district population figures.

Calc: Sum sums up any numeric fields in the table in use.

Dup

Choose *Dup* when adding multiple records in which most of the information is the same. Press <ENTER> on fields holding duplicate data; overwrite data in others.

Edit

SHAMAN allows two levels of editing, *Key Fields* and *Non_Key Fields.*

- For normal editing of a record, choose *Edit Non_Key Field.* This option allows the user to edit non-key fields specific to that record.
- *Key Fields* are rarely edited. Any changes to key fields will affect any child records. USE WITH CAUTION.
- To edit memo fields, choose *Memo* from the action menu.

Find

Find allows the user to search for specific records in the table in view or in a related table. Popup boxes step the user through the *Find* function.

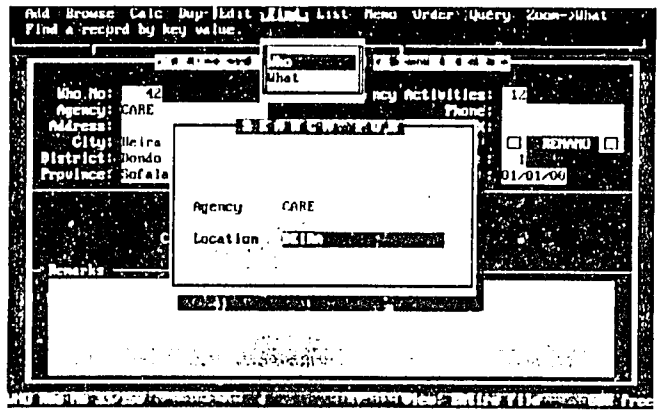


Figure II:8 - Find pulldown menu and dialog box allowing Find on Agency/Location for WHO file.

List

The List menu option lists all reports related to the table in view. Reports can be output to screen, printer, or another file. Popup boxes step the user through the List function.

- See Reports, Part II, Chapter 3.

Memo

SHAMAN allows virtually unlimited text to be included in its memo fields. Choose *Memo* and enter text as needed.

Some records contain memos but they do not appear onscreen. In these cases a √ in an onscreen checkbox indicates a memo. Access by selecting *Memo*.

Some memos appear in the record onscreen, under Remarks or other named memo field, e.g., Objective or information source. The entire memo text may not appear onscreen. It can be accessed by selecting *Memo*.

Some records contain more than one memo field. A pulldown menu under the *Memo* option identifies these choices.

Edit a memo by selecting *Memo* while the related record is in view.

Order

Order allows the user to change the order in which records in a table appear. It is especially useful when using *Browse* to be able to view records in a chosen order.

Query

Query allows the user to select a sub-set of the data in the current or related tables. Subsequent operations, such as *Browse*, or *Calc*, will include only that sub-set. The message View Filtered by query will

appear in the lower right screen.

Query is especially useful to get quick answers to ad hoc enquiries when a formal report is not required.

A number of options appear in a pulldown menu when the *Query* option is selected.

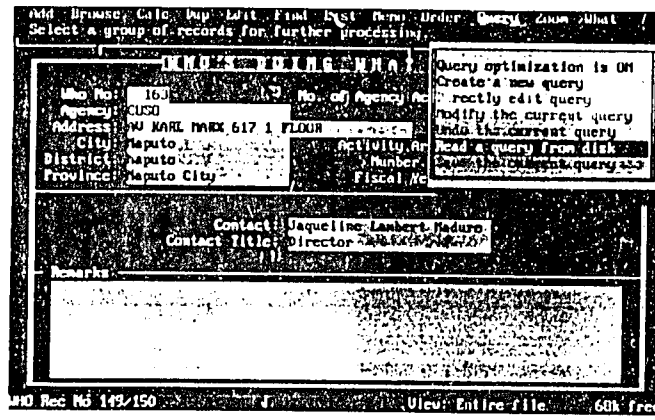


Figure II:9 - Query pulldown menu showing Query options.

- *Query optimization is ON* generally gives the user a faster response time. However, it must be turned off for any queries on numeric fields.

  To toggle between on and off, hit <ENTER >while Query Optimization is highlighted .

- Selecting *Create a new query* pops open a series of windows which step the user through making the query.

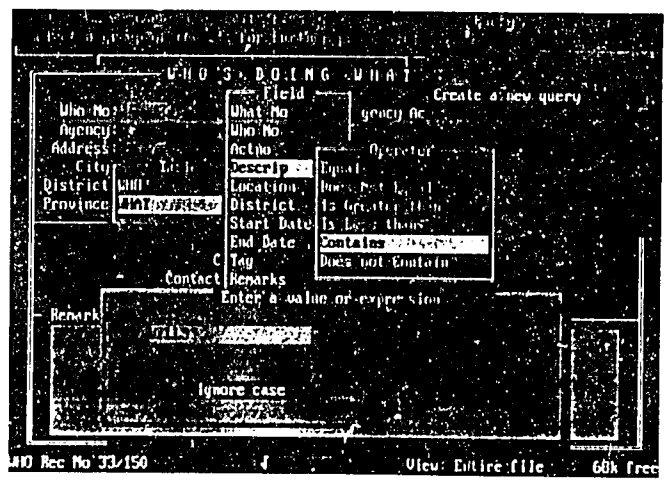


Figure II:10 - A Query on the WHAT file asking for all records where the field Description contains the word "WELLS".

- *Directly edit query* takes the user into a programmer's view of the query. A normal user will not use this option.

- *Modify the current query* reopens the *Create a new query* window.

- *Undo the current query* removes the query filter. The message *View: Entire File* returns to the lower right screen.

- *Read a query from disk* pops up a list of any previously saved queries related to the table in view.

- *Save the current query* allows the user to save frequently used queries, accessing them later with the *Read a query from disk* command.

Zoom

Choose *Zoom* to move from the record in view to one in a related table. You can then do full-screen maintenance on records in the related table.

To "UnZoom" press <ESC>. You'll be taken back to the original screen.

Using On-Line Help

The SHAMAN manual can be accessed on-line by pressing <Shift> <F1>.This action will take you into Expert Help, where help on any subject in the SHAMAN manual can be called up.

If you need help to use On-Line Help, press <F1> when in Expert Help .

Chapter 2

Database Modules

In this chapter, each database module is described and illustrated.

Shaman's database is made up of a number of interrelated modules:

Consolidated Humanitarian Assistance Program
Who's Doing What Where?
Resources
Population
Partisan Dispatches
Incident Reports
Mine Assessment and Survey
Administration

These modules appear to the user to be stand-alone units, but in fact the parts are tightly integrated, sharing files (tables) with other modules. For example, the Location table is shared by nearly all modules, acting as an authority table for spellings in use for towns, districts, etc. This standardizations ensures that valid comparisons and reporting can be done when drawing information from different segments of the database.

In this chapter, each database module is described and illustrated. Data entry tips provide full instructions for data entry in main files. Support files are addressed in Chapter 4, Utility. A glossary of terms specific to the humanitarian assistance data being entered is included below. It will be especially useful for entry of the crucial CHAP data.

Glossary of Humanitarian Assistance Terms

Activity Specific action required to respond to a defined need within a given sector, e.g., "Supply of Agricultural Tools." (See "Category" and "Sector").

Carryover Funds committed within one time frame (i.e., 92/93) which only become available in another (i.e., 93/94).

Category A common programming focus for a set of related sectors, e.g., "Restoration of Essential Services." (See "Sector" and "Activity").

Channel The organization disbursing funds to an implementing agency.

Commitment A firm programme intention on the part of the donor.

Contribution Funds in kind or in cash, made available by a donor for a specific purpose.

Disbursement The actual transfer of funds from the donor to the channel or implementing agency.

Donation The funds that are committed by a donor.

Donor The organization which is the source of the funds and has the decision-making power to direct the resources to the country.

Implementing Agency The organization carrying out the activity.

Obligated Funds Funds in which a signed contract between donor and implementor for a specific activity has been finalized.

Pledge Funds promised by a donor for a country.

Sector A set of related activities which have a common programming focus, e.g., "Agriculture." (See "Category" and "Activity").

Consolidated Humanitarian Assistance Programme (CHAP) Module

The Consolidated Humanitarian Assistance Programme is the heart of the SHAMAN database system.

The CHAP module is the heart of the SHAMAN database. It contains records of all humanitarian assistance requirements identified in a country, pledges made by donors, and $ amounts needed to meet these requirements. Reports generated from CHAP data become the primary humanitarian assistance coordination documents.

NOTE: Clean data collection and entry is critical; rigorous validation procedures must be implemented.

The CHAP module contains two main files:

Activity File
Pledge File

and uses four support files:

Activity Title File
Needs File
Donor File
Agency File

The text in the Activity file amplifies the objective of the activity and summarizes the related situation. It can be of any length, but it is

suggested that for clarity of presentation it be limited to fit a one-page per activity format. (See Appendix D: Sample Reports, CHAP).

NOTE Analyzing and setting up the Activity Title File and the Needs File is a crucial first step in using the CHAP module. How well that is done dictates the usefulness of the CHAP.

Activity File

The Activity file contains the key data on CHAP activities. Activity numbers and titles are determined during the customizing phase. Creating Authority files for these all-important tables is done at setup by the system administrator in consultation with senior management.

Initially, the Activity file requires extensive data and text entry. As the CHAP matures and needs are refined, text may need to be edited. However, the basic categories, sectors, and activity numbers and titles must remain the same. The child Pledge records, other database modules, and integrated SHAMAN reports are all dependent on the Activity file.

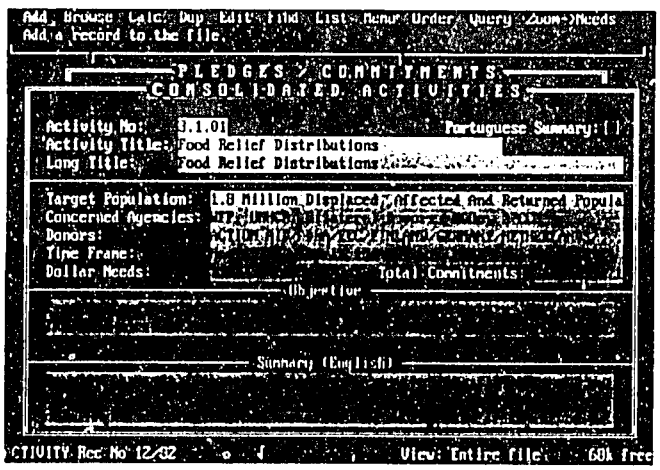


Figure II:11 - Activity file screen.

Note: The Activity file must initially be set up from the *Support* ... module. Select *Utility*, then *Support* ..., then *Activity File Maintenance*. (The Activity Title File must be set up first).

Once the Activity file has been set up, Activity records can be accessed by Zooming from the related Pledge records.

Data entry tips:

- A picklist can be called up from the Activity No. field.
- The Activity Title and Long Title are filled in by SHAMAN when the Activity No. is entered.

- Concerned Agencies are generally implementing agencies. There is often more than one.

- Donors, Dollar Needs, and Total Commitments are information fields only. Information is updated dynamically by SHAMAN.

- To enter the main text select *Memo* and then *Objectives* and *Summary*.

To enter a summary in another language select *Memo* and *Other Lang* If another language summary is entered, a X will appear in the check-box.

(The field title for the [Other language] Summary is changed in the defaults file. For example, in Mozambique the field title is customized as "Portuguese Summary".)

NOTE In most cases, SHAMAN will be used in countries where national data entry staff have English as a second language Careful editing of the lengthy text in this file is recommended

- *Zoom -> Needs* is useful to get a quick view of $ breakdown of Needs. Any modifications should be made by the system administrator in the Needs Support file.

Pledge File

Pledge records are child records of their parent records in the Activity file. There may be several child records for each parent record, i.e., several pledges for each activity.

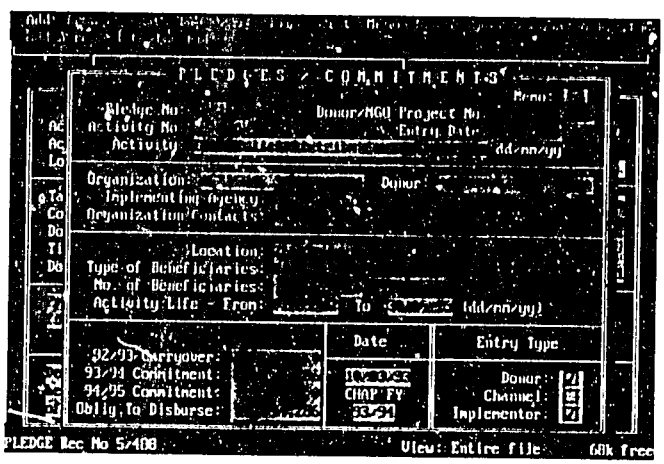


Figure II:12 - Pledge file screen.

Data entry tips:

- The default for Entry Date is the current date. Overwrite if necessary.

- Picklists can be called up from the Activity No., Organization, and Donor fields.

- The Activity title is filled in by SHAMAN.

- The Organization, Donor and Implementing Agency fields cannot be left blank.

- The default value for Implementing Agency is "To be determined." If agency is known, overwrite.

- More than one Location can be entered. As a convention, using the first 3 letters of a location + allows a number of locations to be entered in the field, e.g., "Tet Gaz Map" is entered as the abbreviated provinces of Tete, Gaza, and Maputo.

- The CHAP FY (fiscal year) is entered by SHAMAN. It must be changed in the Defaults file at fiscal year end.

Note The UN Humanitarian Assistance fiscal year runs from 1 May to 30 April

- Under Entry Type, one or more choices must be checked. Use <SPACEBAR> to toggle the X on or off.

- A Donor is the source of the funds, a Channel is a middleman administering the funds, and an Implementor is the organization carrying out the activity. For example, the EEC contributes $200,000 for Water Supply and Sanitation to UNHCR. UNHCR disburses the money to CARE for a well-drilling project. The EEC is the Donor, UNHCR is the Channel, and CARE is the implementor. However, if UNHCR commits the money to their own water projects, they are the implementor. An organization can fit more than one category, depending how the money is used, e.g., UNHCR could fit all three categories if they committed money from their own funds, used part of it for their own projects and gave some to CARE.

NOTE: Because the data for the Pledge file come from varied sources and can be incomplete or ambiguous, verification is often necessary. It is recommended that one person be responsible for verification and standardization of data.

Who's Doing What Where Module

The Who's Doing What Where Module documents the humanitarian assistance activities currently being implemented by agencies active in the field. It is useful for coordination of always limited resources.

The Who What Where Module contains two main files:

Who File
What File

and uses three support files:

Location File
Agency File
Activity Title File

Who File

The Who file holds basic information on humanitarian assistance agencies active in the country. These agency records are the parent records for all related What records. An agency should have one Who record for each field office it maintains.

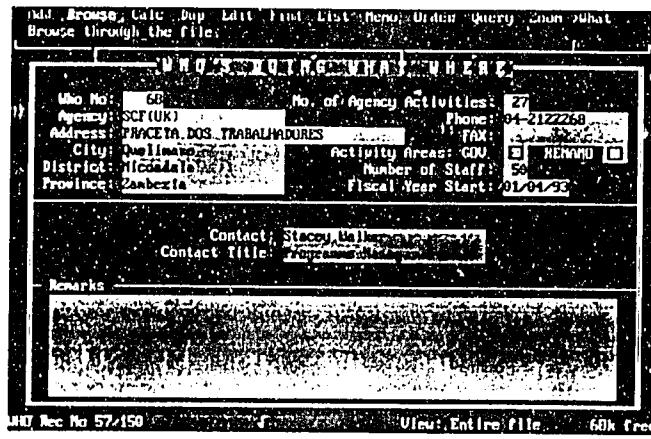


Figure II:13 - Who file screen.

Data entry tips:

- Picklists can be called up from the Agency and City fields.
- The No. of Activities will be filled in by SHAMAN as child What records are entered.
- District and Province are filled in by SHAMAN when City is entered.
- District is not necessary for address purposes, but useful for information.
- In Activity Areas use <SPACEBAR> to toggle √ on or off.
- Select *Memo* to enter any remarks.
- Select *Zoom -> What* to enter child records in the related What file.

What File

Each Who record can have one or more dependent What records. The What File holds these child records. Each What record documents an

activity coordinated by the related Who agency.

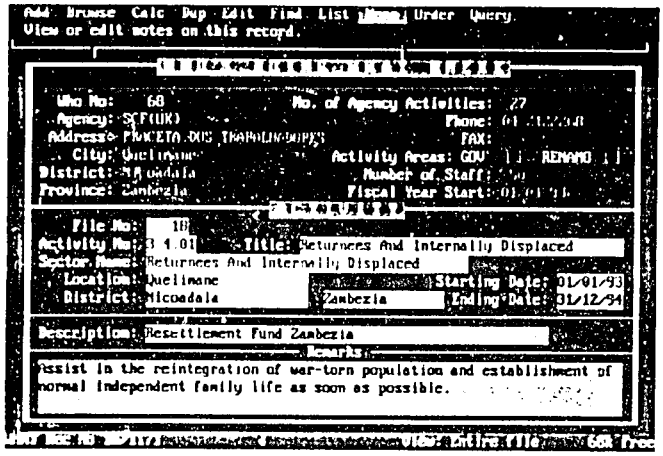


Figure II:14 - Child What file record overlaid on Who screen.

Data entry tips:

- Parent record information appears in the top of each child record.
- Picklists can be called up from the Activity No., Location, and District fields.
- The Location field can be left blank. To access the picklist, press any key + <ENTER>.
- In the Activity No. picklist, choose the Activity No. that most closely matches the activity described. For example, if an agency describes their activity as "supply of hoes and shovels", choose *4.1.03 Agricultural Tools*. If the agency describes their activity as "agricultural support", choose the broader sectoral number *4.1 Agriculture*.
- The actual Agency's description of their activity, e.g. "supply of hoes and shovels," should be entered in the Description field.
- Select *Memo* to enter any remarks.

Note: Data collection in a standardized format is essential if this module is to be useful. A suggested data-collection form can be found under *List*.

Resources Module

The Resources module contains information on a country's resources in various sectors, e.g., health, education, water. Reports from data in this module are useful planning documents. The addition of mapping software makes this an even more powerful analytical tool.

The Resources module contains one main file:

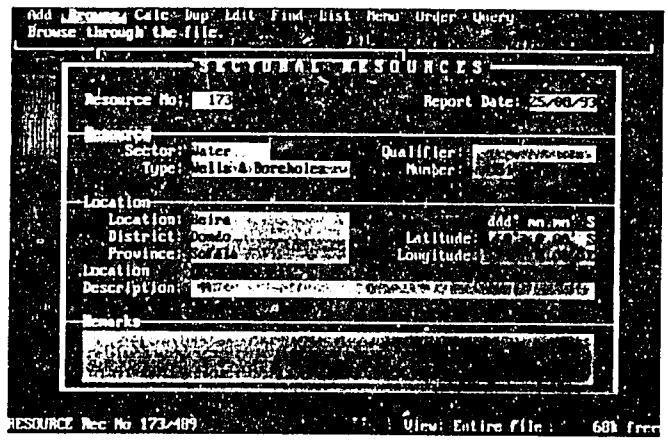
Resources File

and uses two support files:

Location File
Resource Type File

Resources File

The Resources file holds information on the sector, the type and number of resources, and their location. Careful analysis of information needs before customizing this file can make it useful in a variety of situations.



Figuref II:15 - Resources file screen.

Data entry tips:

- The default for Report Date is the current date. Overwrite with Report Date if available.
- Picklists can be called up from Sector, Type, Location, and District fields.
- The Location field can be left blank. To access the picklist, press any key + <ENTER>.
- Province is filled in by SHAMAN.
- The Qualifier field is used to describe the resource being counted. It can be used in various ways. For example, if the Resource Sector is Health and the Resource Type is Hospitals, the Qualifier could be "Number of Beds," or it might be the type of hospital, e.g., "Provincial Hospital."
- The Number must be greater than 0.
- Location Description helps to physically locate resources within

an area, e.g., "Near the bridge approximately half way between Mocubela and Maganja."

- The Latitude and Longitude can be filled in to pinpoint the resource location. This is especially useful if mapping software is being used in conjunction with SHAMAN data. If Lat. Long. is not entered, SHAMAN will select the coordinates of the nearest town or district seat in any reports or maps.

- Enter Latitude and Longitude as degrees and decimal minutes, e.g., a latitude of 20° 10' 30" is entered into the field as "20 10.5 S."

Incident Reports Module

The Incident Reports module provides a means to track incidents affecting humanitarian assistance activities, e.g., theft of food aid, vehicle accidents, harassment, or land mines, and relate clusters of incidents to specific locations or activities.

The Incident Report module contains one main file:

Incident File

and uses one support file:

Location File

Incident File

The Incident File allows entry of the details of various types of incidents in a series of memo fields. Precise locations can be pinpointed. Regularly produced incident reports can be a useful security measure. With the addition of mapping software this module becomes an even more powerful analytical tool.

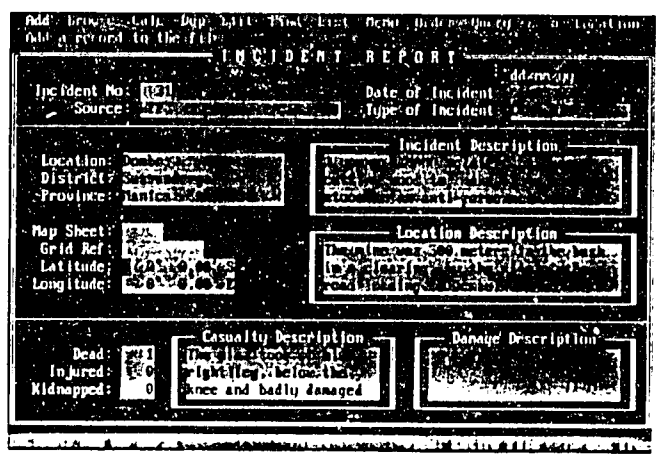


Figure II:16 - Incident file screen.

Data entry tips:

- The default for Date of Entry is the current date. Overwrite with date of incident or incident report if available
- Picklists can be called up from Type of Incident, Location, and District fields.
- Location may be left blank. To access the picklist, press any key + <ENTER>.
- Province is filled in by SHAMAN
- Location Description helps locate the incident within an area, e.g., "Near the bridge approximately half way between Mocubela and Maganja."
- The Latitude and Longitude can be filled in to pinpoint the incident location. This is especially useful if mapping software is being used in conjunction with SHAMAN data and also for pinpointing Mine incidents. If Lat. Long. is not entered, SHAMAN will select the coordinates of the nearest town or district seat in any reports or maps.
- Enter Latitude and Longitude as degrees and decimal minutes, e.g., a latitude of 20° 10' 30" is entered into the field as "20 10.5 S."
- Select *Memo -> Incident Description* to describe the basic incident.
- Fill in other memo fields as appropriate.
- *Zoom -> Location* if you need to enter a location that does not appear on the picklist.

Note: Check the picklist carefully before entering a new location to ensure you are not just using an alternate spelling. Your entry will become part of the Location Authority File.

Population Module

The Population file holds population figures by district from various sources. It includes figures on the number of persons currently receiving humanitarian assistance benefits.

The Population module contains one main file: Population File

and uses one support file: Location File

The source field names for population figures are customized at setup. There are 5 fields for population estimates from government or non-

government agencies and 2 fields for the latest government census figures.

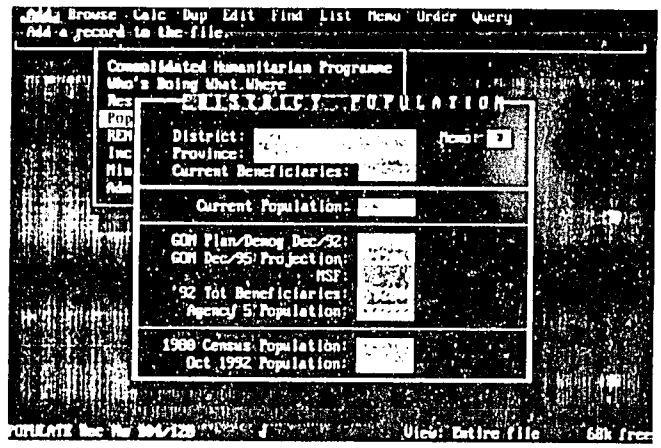


Figure II:17 - District Population file screen.

Note Analyze population source fields carefully before setup. They cannot be changed later without changing all dependent R&R Reports.

Population File

Data Entry Tips:

- A picklist can be called up from District.
- The Province is entered by SHAMAN when the record is saved.
- Enter population figures as appropriate.

Partisan Module

The Partisan module holds information on dispatches of food and non-food items to areas under Partisan control.

The Partisan module contains one main file:

Partisan File

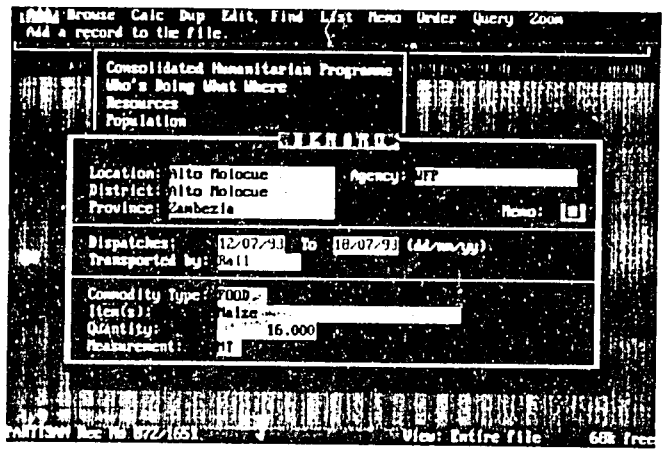
and uses three support files:

Location File
Agency File
Mode of Transport File

The fields name of the Partisan group is customized at setup, e.g. "RENAMO" in Mozambique, or "SPLA" in Sudan.

Partisan File

Figure II:18 - Partisan file screen.



Data Entry Tips:

- Picklists can be called up from Location, Agency, and Transported By fields.
- District and Province are entered by SHAMAN when Location is filled in.
- Dispatches is set up to record dispatches between any two dates.

Note There is an important difference between dispatches and deliveries. Especially in war-torn situations the actual delivery amounts may not be known.

- The default for Transported By is TRUCK. To accept, press <ENTER>. To change, type <x> and press <ENTER> to access picklist.
- Commodity Type and Measurement are dropfields.
- The default for Commodity Type is FOOD. To accept, press <ENTER>. To change, press <SPACEBAR> to access list. (R&S = Relief and Survival).
- Item is the type of commodity, e.g., "maize", or "plastic sheets."
- The Number must be greater than 0.
- The default for Measurement is MT (metric tons). To accept, press <ENTER>. To change, press <SPACEBAR> to access list.

Mine Assessment and Survey Module

The Mine Assessment and Survey module provides a means to track mine locations as they are reported, found, and removed. It can be used by de-mining teams and is also a valuable source of information for humanitarian assistance agencies working in areas mined or suspected to be mined. With the addition of mapping software, known dangerous areas can be quickly identified.

The Mine Assessment module contains two main files:

Mine Assessment File
Mines File

and uses the following support files:

Location File
Mine Types File

Mine Assessment File

The Mine Assessment file contains records of mines reported or found. Information can be general or specific. Memo fields allow extensive text.

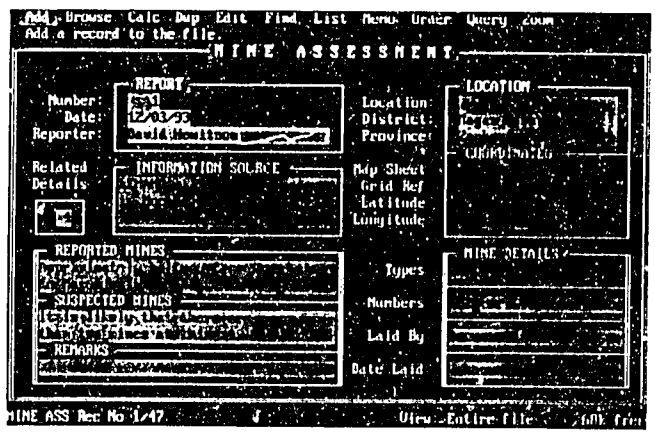


Figure II:19 - Mine Assessment and Survey file screen.

Data entry tips:

- The default for Date is the current date. Overwrite with report-date if available.
- Reporter can be a named individual or title, e.g. UNOHAC Rep., Nampula.
- Picklists can be called up from Location and District fields.
- Location may be left blank. To access the picklist, press any key + <ENTER>.

- Province is filled in by SHAMAN when record is saved.
- Number of Related Details is filled in by SHAMAN as child records are entered.
- The Latitude and Longitude can be filled in to pinpoint the mine location. This is especially useful if mapping software is being used in conjunction with SHAMAN data and also for pinpointing mine locations. If Lat. Long. is not entered, SHAMAN will select the coordinates of the nearest town or district seat in any reports or maps.
- Enter Latitude and Longitude as degrees and decimal minutes, e.g., a latitude of 20° 10' 30" is entered into the field as "20 10.5 S."
- Mine Details are general at this level. More than one Type can be entered, e.g., "anti-tank, anti-personnel" or "AT, AP."
- Number is approximate. A qualifying note can be added, e.g. "unknown," or, "I seen; probably more."
- Access text fields by selecting *Memo* and then relevant field: *Source, Reported, Suspected, or Remarks.*
- *Zoom -> Mines* for child records of the Mine Assessment records.
- *Zoom -> Location* if you need to enter a location that does not appear on the picklist.

Note: Check the picklist carefully before entering a new location to ensure you are not just using an alternate spelling. Your entry will become part of the Location Authority File.

- *Zoom -> Mine* types to enter new types of mines found or reported.

Note: Browse the Mine Type file before entering a new mine type to ensure you are not just using a variant. Your entry will become part of the Mine Types Authority File.

Mines File

The Mines file is the only module in SHAMAN in which more than one record appears on a screen. All data related to one Mine Assessment record are immediately accessible to the user in a browse-like screen.

The Mines records provide specific information on the actual mines found and cleared in the area reported in the assessment.

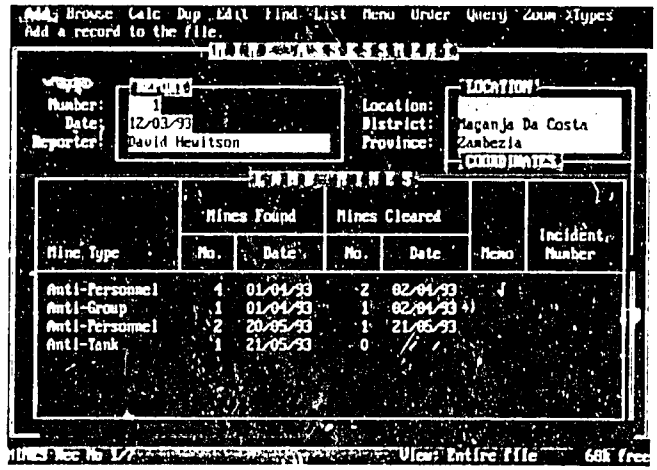


Figure II:20 - Child Mines file record overlaid on Mine Assessment screen.

Data entry tips:

- Append a new record line by selecting *Add*.
- Edit old records by selecting *Edit*. Then scroll to highlight the record line and press <ENTER>.
- A picklist can be called up from Mine Type.
- Some Mines records will relate to incidents reported in the Incident file. However SHAMAN does not relate the records automatically. To look for a related Incident Report number, *Zoom* -> *Incident* and use *Find* or *Browse* to locate the
- *Zoom* -> *Mine* types to enter new types of mines found.

Note: Browse the Mine types file before entering a new Mine Type to ensure you are not just using a variant. Your entry will become part of the Mine Type Authority File.

Administration Module

The administration module can be modified to add files useful to an organization, such as a Personnel File. This version of SHAMAN contains one file, a mailing list, that makes it easy to disseminate SHAMAN reports or other publications, notices, etc. The module allows up to eight document types to be entered. Names and addresses of individuals and organizations are entered once. Mailing labels specific to the document to be sent are produced as needed.

The Administration module contains one main file: Mailing Labels

and uses one support file: Classify File.

Mailing List File

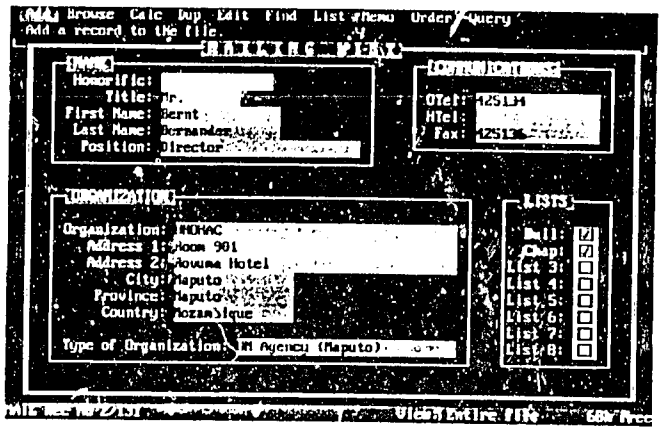


Figure II:21 - Mailing List file screen.

Data entry tips:

- The upper Name fields can be left blank, and only the Organization name and address filled in.
- Picklists can be called up from Honorific and Title. Press any key and <ENTER> to access, or leave blank by pressing <ENTER>.
- The default on Title is Mr. Press <ENTER> to accept, <SPACEBAR> to override with another choice.
- A picklist can be called up from Type of Organization.
- Under Lists, scroll the list of documents to highlight your choice. Use <SPACEBAR> to toggle the √ on or off. Any number of the documents listed can be selected.
- Mailing labels for a particular document will be produced only for those records containing a √ in the relevant checkbox.

Chapter 3

SHAMAN Report

The pulldown menu under Report gives access to:

Reports
Printer Support
Prepare Memos for Reports
Update Activity File for Reports

Reports

SHAMAN reports are created using R&R Report Writer® and these report "shells" are then integrated into SHAMAN using the Report Manager Utility. Most standard reports are listed in a pulldown menu under *Reports*. The list will change as further reports are developed and integrated into SHAMAN.

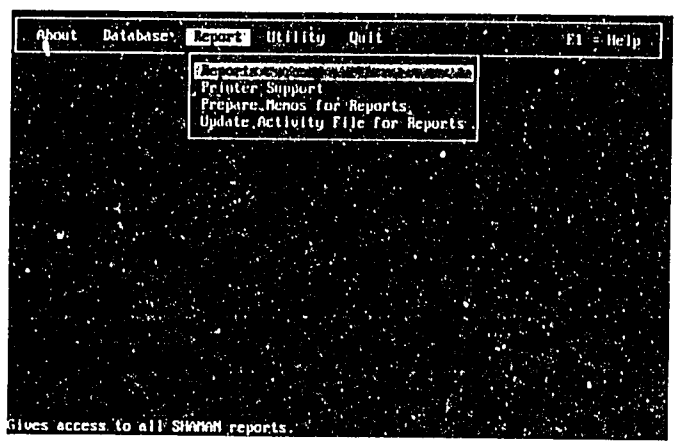


Figure II:22 - Pulldown Report menu.

Reports can be accessed from this menu and output to the screen, the printer, or another file. (The same reports can be accessed under the *List* menu when working with files. Only the reports related to the file being used appear in its List).

To select and output a report:

1 Scroll to highlight the report and press <ENTER>. A Report Settings menu will pop up with "Go! Print the Report" highlighted.

2 Press <ENTER> to output the report as designated in the Report Manager. (Most SHAMAN reports are set to output to screen. The output setting can be changed in the Report Manager Utility).

OR: to override the designated setting, change Destination (Screen, Printer, or File), set Scope and Copies as required. Then select "Go! Print the Report".

Note: It is easy to output a report to printer from the Reports menu choice, but the report will be put out for a generic printer. For higher quality printing on a specific printer, e.g. an HP LaserJet, shell out to R&R Report Writer® under *Utility -> External*

See also Prepare Memos for Reports and Update Activity File for Reports below.

Printer Support

Printer Support is configured for either a generic printer or laser printer at system setup. The Printer Support option is rarely needed by normal users of SHAMAN. One possible use is to enhance the appearance of a <Print Screen> by changing the number of dots per inch. The system administrator may need to use it from time to time, especially if a non-standard printer is being used.

Note: This option is not used to change the printing characteristics of Reports.

Prepare Memos for Reports

Use the Prepare Memos for Reports option before shelling out to R&R Report Writer® to print a report that contains memos, e.g. the CHAP. This option strips out the soft carriage returns which otherwise print as box symbols in R&R. SHAMAN does this automatically for reports output to printer from within SHAMAN. (See Reports above).

Shell out to R&R Report Writer under *Utility -> External.*

Update Activity File for Reports

Use the Update Activity File for Reports option before shelling out to R&R Report Writer® to print any report that will use the donor field in the Activity file, e.g., the CHAP. This option takes all donor information from the Pledge files and updates the donor field in the Activity file. SHAMAN does this automatically for reports output to printer from within SHAMAN. (See Reports above).

Shell out to R&R Report Writer under *Utility -> External*

Chapter 4

Utility

The SHAMAN Utility menu is divided into four sections:

System
Support Files
Utilities
External

Novice users are locked out of most of these modules. Only the system administrator should have access to the data dictionary and other modules which directly affect SHAMAN's operation.

System

System modules relate directly to administration of the SHAMAN program.

Choose *System* to access:

SHAMAN Data Dictionary
Help Index
Query Manager
System Administrator
Report Manager

SHAMAN Data Dictionary

The Data Dictionary is the manager of the SHAMAN program. It tracks information about all files, indexes, relations, and commands in SHAMAN.

WARNING Only the system administrator has access to the Data Dictionary. Complete instructions for its use and maintenance can be found in the Artful Two User's Guide. This is definitely not an area for novice programmers.

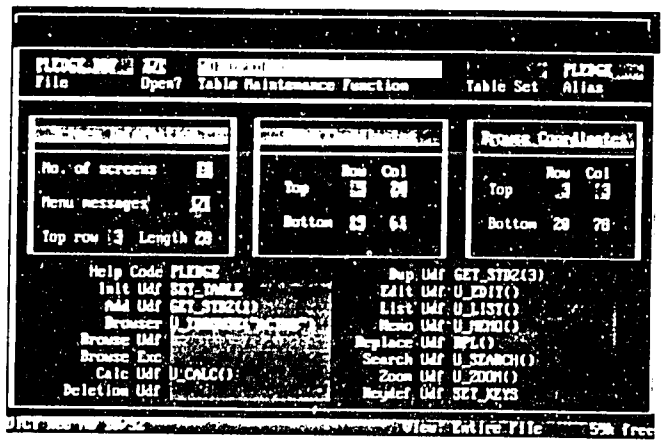


Figure II:23 - Data Dictionary screen showing Pledge file record.

Use the Data Dictionary to:

- Add a new main table or edit an existing main table.
- Add a new index or change a table's index.
- Add or change a table's relationships to other files.
- Add or change a table's Zooms to other files.

1 To see any dictionary record, move the light bar to it and press <ENTER>. There are several dictionary record screens. <PgUp> and <PgDn> move you through them.

2 To add a dictionary record (table) press <INSERT>.

3 To delete a file press <DEL>.

Help Index

The Help Index lists and gives access to all the context-sensitive Help screens in SHAMAN.

Use the Help Index to:

- Get a quick overview of what Help exists.
- Read through the Help files.
- Edit the Help screen.
- Edit the Help Key, the Read Variable and the Coordinates of current Help record.

To Edit the Help Screen:

1. Scroll to the Help record you want and press <ENTER>.
2. Edit the text and press <Ctrl-W> to save changes.

- The system administrator can also edit Help screens as they are called up. It may be useful to add some extra Help from time to time for specific problems or to provide Mission-specific examples.

To Edit the Help Key, the Read Variable and the Coordinates:

1. Scroll to the Help record you want and press < F2 >.

2. Edit selected fields and press < Ctrl-W > to save changes.

- Normally, this editing only involves the name of the Help screen and rarely, the Help Key.

Query Manager

The Query Manager allows direct access to the Saved Queries table. The existing queries can then be edited directly in the file.

This function would normally be used only to delete queries if the number of queries saved to disk became unmanageable.

Use the Query Manager table to:

- Delete a Query saved to disk.
- Modify a Query.
- Change the name of a Query.

To delete a Query:

1. Select *Query Manager*.
2. Select the Query you want to delete. This can be done using the *Browse* or *Find* actions or the Arrow keys.
3. Press <DEL>.

To modify a Query:

1. Select *Query Manager*.
2. Select the Query as above.
3. Press *Memo*. The Query can then be modified in the Memo box.

To change the name of a Query:

1. Select the Query as above.
2. Select *Edit* and then *Edit all Fields*.
3. Change the name of the Query in the Query Name field.

System Manager

The System Manager allows the user to change colors and passwords.

- To change colors, select *System Manager*. Then select and *Edit Non_key Fields*.

- To add or change passwords, select *System Manager*. Then select *Edit* and *Edit Key Fields*.

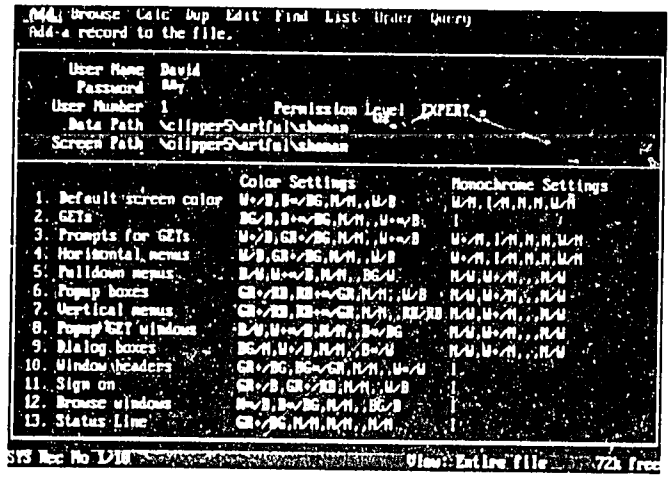


Figure II:24 - System Manager screen used to change user passwords, levels and colors.

Report Manager

The Report Manager utility allows externally created reports from R&R Report Writer® to be integrated into SHAMAN. It creates a path between SHAMAN and the Report Libraries. The Reports can then be accessed under the Report or List menu choices.

To add a new report to SHAMAN:

1. Select *Report Manager* and go to Screen 1.

2. Enter the name of the report and the commonly-required settings for query, index, scope, copy, and destination.

These values become the default settings, allowing the user to quickly access the report under *Report* or *List* and select "Go! Print the Report." The values can be temporarily overwritten within the Reports/List menu if needed.

3. Go to Screen 2 with <PgDn>.

4. Enter Expert Control Settings for Type of Report, its Library and other technical information.

- Field level help is available by pressing <F1>, but for full information, refer to the Artful Two User's Guide.

Support Files

Support files are the authority files used by SHAMAN. Support files can be related to more than one database module. They ensure use of standard names, spellings, acronyms, etc. Only names from a related support file will appear in a field's picklist. For example, the picklist for Agency will contain only the list of agencies entered into the Agency file.

Careful analysis and ongoing maintenance of the Support files is crucial for the smooth running of SHAMAN. Much of the initial work is done at system setup.

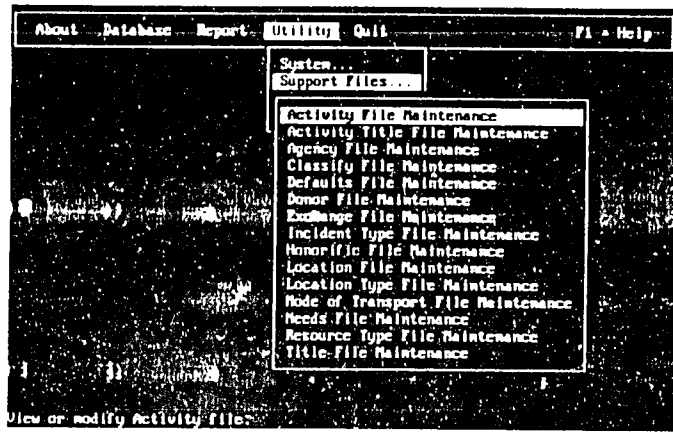


Figure II:25 - Support Files pulldown menu.

Use Support Files Maintenance to:

- Customize SHAMAN
- Add or modify items in a support file.
- Get information on a support file.

At setup, SHAMAN's authority files must be customized for a particular country or situation. This will include Donors, Activity Name and Numbers, and some field names. Support Files Maintenance is used to create these authority files.

From time to time, modifications need to be made. For example:

The CHAP FY (fiscal year) is in SHAMAN as a default value. To change the value at the start of a new fiscal year, go to *Utility -> Support Files -> Defaults File Maintenance* and enter the new fiscal year.

Or:

If a new donor pledges funds, the Donor field in the Pledge file will not accept the new donor name until it is entered into the Donor authority file. Go to *Utility -> Support Files -> Donor File Maintenance* to add the new Donor name.

Note: Ensure that any additions are not simply variants of a name already in an authority table. Browsing the relevant file using the Support Files maintenance option is an easy way to check.

Activity Title File Maintenance

The Activity Title file is the first file to be customized when setting up SHAMAN and is vital to the operation of the SHAMAN system. It

could be considered as a table of contents to the CHAP. The number identifies the category, the sector, and the activity. The title describes the activity in a short and long form:

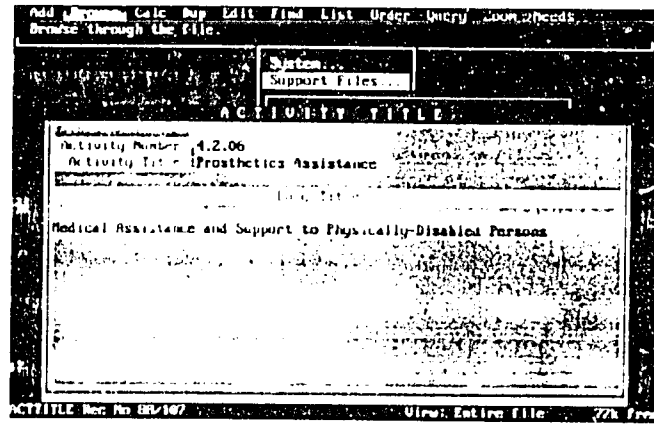


Figure II:26 - Activity Title file maintenance screen.

In Activity No. 4.2.06:

4 represents the category Restoration of Essential Services
.2 represents the sector Health
.06 represents the Activity Prosthetics Assistance

'Prosthetics Assistance' is the short title. The long title for the activity is 'Medical Assistance and Support to Physically Disabled Persons.'

The short and long titles may be the same.

Two Activity numbers are reserved for general area activities. The number ".00" represents a general allocation. For example, in Activity No. 4.2.00, ".00" would represent money slated by a donor for the Health sector without specifying an activity.

The number ".99" represents donor unallocated, but UNOHAC reserved funds. For example, in Activity No. 4.2.99, the ".99" would represent unallocated funds reserved by UNOHAC for the Health sector.

These numbers are intended only as a "catch all" for funds that have not yet been allocated to specific activities or which are designated by the donor for use anywhere within a program, sector, or activity. They show up in the financial summary report, but not in the activity summary report.

Note The Activity Title file cannot be modified after setup without serious consequences for the integrity of SHAMAN. It is possible to add a new activity to a sector, or even a new sector to a category, as long as they are not inserted into a run of existing numbers. However, changing numbers or titles, or even shifting them will affect every related file and report.

Agency File Maintenance

The Agency file contains Agency names, type of agency, short and long descriptions, e.g., Agency Title: CIDA; Long title: Canadian International Development Agency; Type: BIL (Bilateral).

Customize the file to include all the agencies active in a country or area, or only a specific list. New agencies can be added as needed.

Classify File Maintenance

The Classify file is the authority table for mailing list classifications. Use it to classify the individual mailing list record as to type of organization represented. For example, DON, Donor; NGO, Non-government organization; EMB, Embassy. This allows one to print out a mailing list of just Donors or just NGOs.

Defaults File Maintenance

Defaults File Maintenance is the utility used to customize SHAMAN at setup and to modify SHAMAN default settings as required.

The Defaults file holds information customized for particular field names, and in some cases, field contents.

For example, one country might use the term Province, another might use State. The field name can be changed using Edit in *Defaults File Maintenance*.

The Defaults file includes standard names used in R&R Report fields, e.g., Country Header.

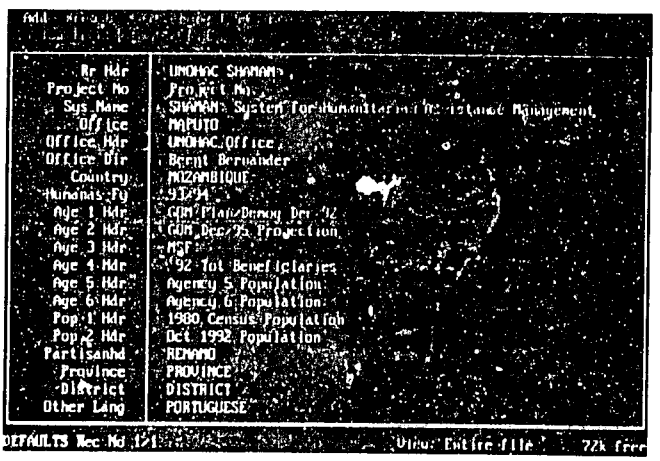


Figure II:27 - Defaults file maintenance screen, page 1.

There are many Default settings. Use <PgDn> to access the second screen.

Donor File Maintenance

The Donor file contains a code name and a description for each Donor, e.g., "USAID," "United States Agency for International Development."

New Donor names can be added as required.

Honorific File Maintenance

The Honorific file contains special titles used to address government officials, or other dignitaries, e.g., "Her Excellency."

Customize as required for your mailing list.

Incident Type File Maintenance

The Incident Type file contains short and long descriptions of incident types, e.g., theft, harassment, land mine. In most cases, the short and long names will be the same.

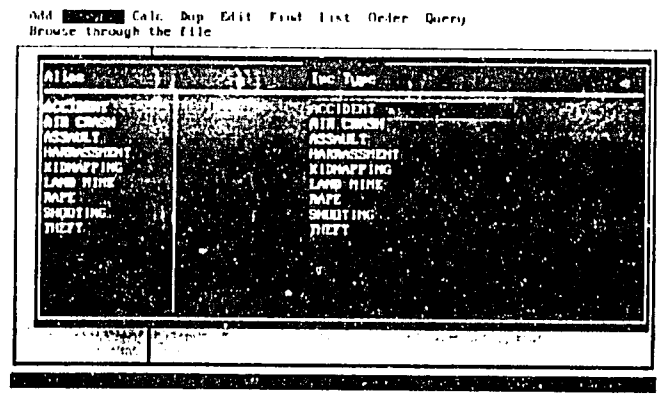


Figure II:28 - Browse screen for Incident Type file.

Location File Maintenance

The Location file contains the Location Name, District, Province, Type of place, Latitude, Longitude, and Location Remarks.

In some countries, deciding on standard spelling of place names is difficult, e.g., when transliterating from a foreign alphabet to a Roman alphabet. It is imperative that only one name be used for a single place.

Latitude and Longitude must be entered for any location to be pinpointed on a map when using mapping software.

Note In some modules latitude and longitude can also be entered into records e.g. location of land mines. These specific latitude and longitude entries do not become part of the Location file.

- Enter latitude and longitude as degrees and decimal minutes, e.g., a latitude of 20° 10' 30" is entered into the field as "20 10.5 S."

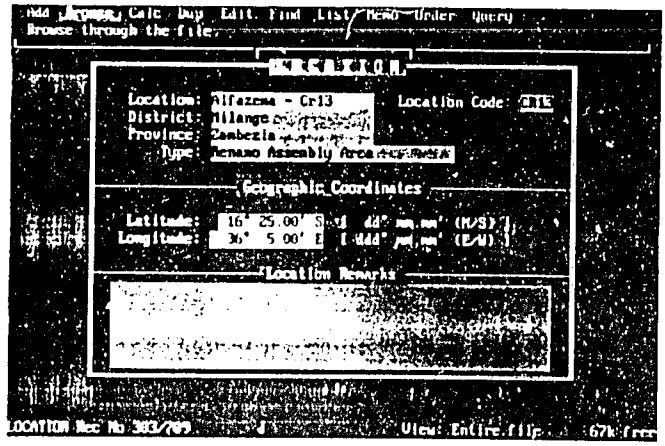


Figure II:29 - Location file maintenance screen.

Using a Location Code is useful if you need to break the location type information down further. For example, in Mozambique, Assembly Areas for Demobilization were designated as NR01, NG12, etc. The CR13 in the above illustration stands for Central RENAMO #13.

Note: For ease of use and because all locations needed cannot be foreseen at setup, some modules allow the user to Zoom -> Location to enter a new location. It is recommended that the system administrator review these new entries from time to time to ensure use of standardized spellings, etc.

Location Type File Maintenance

The Location Type file contains short and long descriptions of location types, e.g., AA, Assembly Area; DS, District Seat; PC, Provincial Capital.

Mine Type File Maintenance

The Mine Type file contains the names and descriptions of mine types, e.g., UXO, Unexploded ordinance; Anti-tank, Anti-tank; AP:Ru T-46, Anti-personnel - Russian T 46; IED, Improvised Explosive Device.

New mine types can also be entered directly from the Mines file by using *Zoom -> Mine Type*.

Mode of Transport File Maintenance

The Mode of Transport File contains the name and code letter for types of in-country transport used for moving commodities, e.g. Truck, T.

Needs File Maintenance

The Needs file is closely related to the Activity Title file. There can be more than one child Needs record for each parent Activity Title record. The Needs file contains the Activity Number and Title, along with the Needs amount in US$.

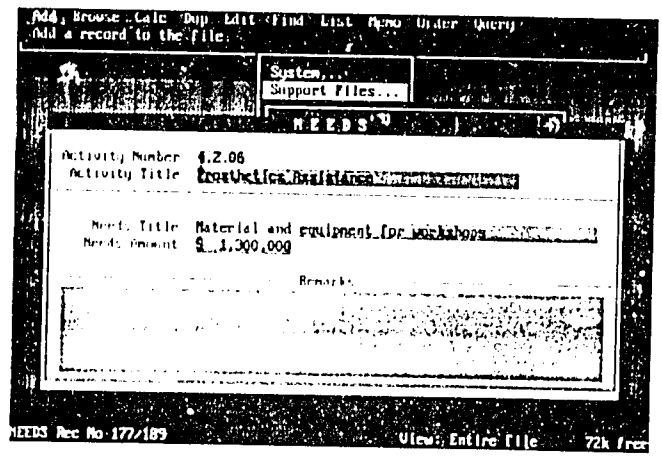


Figure II:30 - Needs file maintenance screen.

Setting up the Needs file is an important part of initial SHAMAN customization. New records can be added as required.

Resource Type File Maintenance

The Resource Type file contains the names of resource types, e.g., wells, schools, their sectors, e.g., water, education, and any remarks.

Determining which resource sectors and types will be tracked is part of the customization process. Additions can be made if required.

Title File Maintenance

The Title file contains name titles, e.g., Mr., Mrs., Sr., used in the title field of the Mailing List.

Utilities

Choose Utilities to perform these actions on SHAMAN files:

Reindex SHAMAN Files
Pack SHAMAN Files
Backup SHAMAN Files
Reindex SHAMAN Files

Reindex SHAMAN Files

The Reindex SHAMAN Files utility reindexes corrupted files. If

there seems to be a problem finding information that you know is in the file, there may be a problem with the index. Try the solution of reindexing the file.

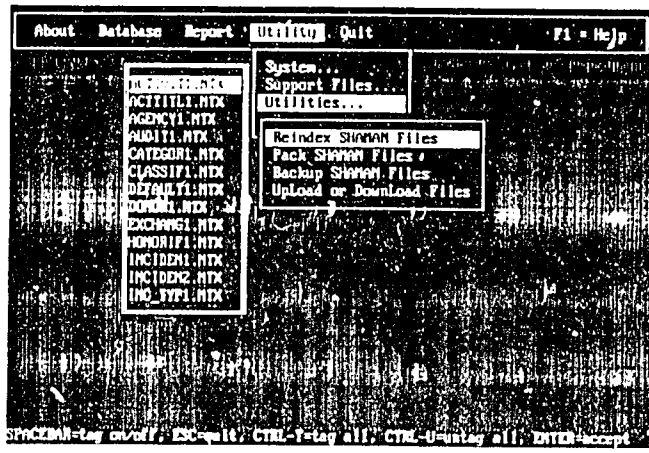


Figure II:31 - Reindex utility showing index picklist.

To Reindex a file:

1. Scroll the list of files and use <SPACEBAR> to tag each file that you want reindexed.

- To tag all the files press <Ctrl-T>.
- To untag all files press <Ctrl-U>.

2. After tagging, press <ENTER>.

- A dialog box will appear with choices to continue [OK] or cancel [Cancel] all reindexing.

3. Scroll to your choice and press <ENTER>.

Pack SHAMAN Files

Packing a file physically removes any records that have been deleted (before packing, deleted records are still in the file even though you can't see them). Packing helps to make the file shorter and will also reindex at the same time.

> WARNING: It may take a long time to pack, especially if the file is big. There is danger of file corruption if the procedure is interrupted for any reason. Use with care.

To Pack selected files:

1. Scroll the list of files and use <SPACEBAR> to tag each file that you want packed.

- To tag all files press <Ctrl-T>.

- To untag all files press <Ctrl-U>.

2. After tagging press <ENTER>.

- A dialog box will appear with choices to either continue [OK] or cancel [Cancel] all packing operations.

3. Scroll to your choice and press <ENTER>.

Backup SHAMAN Files

SHAMAN Backup procedures are fast and simple. Files are first compressed using PKZIP. You are then stepped through the backup process with onscreen choices.

To Backup files:

1. Scroll to "Backup all SHAMAN Files" and press <ENTER>. A series of screens will pop up to do the following:

Screen 1: Select backup drive (Normally choose your 3 1/2" floppy drive).

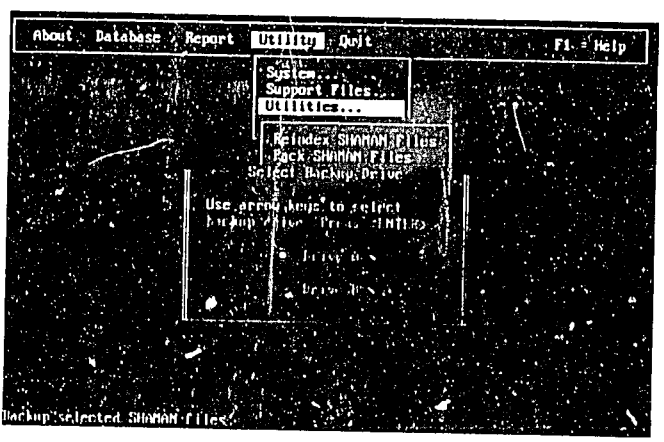


Figure II:32 - Screen 1 of SHAMAN Backup utility.

Screen 2: Compress files with PKZIP and Continue (Press <F1> for information on PKZIP).

Screen 3: Delete old version of SHAMAN.ZIP (Normally OK).

Screen 4: Insert Backup disk (Use high density 1.44mb. 3 1/2" disk).

Screen 5: Delete old SHAMAN.ZIP file on disk if it exists (Normally OK).

Screen 6: Compare size of file and size of free space on disk (If enough space, continue).

Screen 7: Confirm OK to continue.

Screen 8: Confirms successful copy.

To Restore All SHAMAN Files from Backup Disk:

1. Place SHAMAN Backup disk in A: or B: drive and press <ENTER>. If you accept OK, ALL files will be replaced including data files.

To Restore a single file:

If, for example, you want to restore the Location files.

1. Exit SHAMAN to DOS.
2. From DOS type: PKUNZIP A:SHAMAN.ZIP LOCATION.*

External

External allows direct access to:

DOS
R&R Report Writer®

DOS Gateway

DOS Gateway allows quick entry to DOS without exiting SHAMAN. To shell out to DOS:

1. Select DOS Gateway from the Utility Menu..
2. Perform any activity in DOS.
3. To return to SHAMAN, type EXIT on the DOS Command line and press <ENTER>.

WARNING: Use DOS Gateway with caution. The SHAMAN files are open and can be corrupted.

R&R Report Writer

R&R Report Writer® allows quick entry to R&R Report Writer® without exiting SHAMAN. To shell out to R&R Report Writer®:

1. Select R&R Report Writer® from the Utility Menu.
2. Perform any activity in R&R Report Writer®.
3. To return to SHAMAN, type EXIT on the DOS Command line and press <ENTER>.

Chapter 5

Generating Reports

SHAMAN makes it simple to produce up-to-the-minute reports with a few keystrokes. A suite of commonly called-for reports is incorporated into each module. They can be called up on screen, printed, or exported to a text file. These reports are designed using R&R Report Writer®, integrated into SHAMAN, and then generated as needed. New reports can be designed and integrated as required. For information on incorporating and accessing reports, see Part II, Chapter 3, Reports.

This section provides general information on generating new reports, and needed references for the use of R&R Report Writer®.

Generating New SHAMAN Reports

Follow R&R directions for full instructions. The program includes extensive On-line Help accessible with <F1>. See also Using R&R Report Writer® below.

SHAMAN makes it simple to produce up-to-the-minute reports with a few keystrokes.New reports can be designed with R&R Report Writer® and integrated into the system as required.

All SHAMAN reports are kept in a SHAMAN report library:

SHAMAN.RP1 or SHAMAN.RP5 (Windows version)

An existing "SHAMAN Report Skeleton" can be used as a starter for your new report. It already contains Headings, Titles and Boxes which you can then fill in.

Using R&R Report Writer®

Go to R&R Report Writer® (DOS xBase edition) by shelling out from SHAMAN under *Utility -> External* or from DOS. Access R&R Report Writer® for Windows by exiting SHAMAN and entering Windows.

Concentric System's R&R Report Writer® is a powerful report writing program that can be mastered by any reasonably experienced computer user. It has the ability to relate and sort data from many different tables with ease.

"In order to create a report, you select the database file(s) to use. Then you create a report definition that tells R&R how to present the data from the selected files.

This report definition includes a report layout that you design on your screen by typing text where you want it to appear and indicating where data from your files should appear or print. As you create the layout, you can move text and data freely around the report screen.

At any point in designing a report, you can print the report, preview it on your screen, modify it, and save the report output in a disk file. When you've completed the report definition, you can also save the definition so that you can run or modify the report at any time." **R&R Report Writer User's Guide**, p. 23

For complete information on the use of R&R, see:

R&R Report Writer User's Guide. Concentric Data Systems, Inc., Westborough, MA, 1990.

To easily integrate lines, boxes and graphics into your reports, use R&R Report Writer® for Windows.

R&R Report Writer® for Windows

Using R&R Report Writer® for Windows rather than the DOS version of R&R has a number of advantages:

- Graphics and shading can be incorporated
- The drawing of lines and boxes is much easier.
- You have generally better control over most aspects.

The only disadvantage in using R&R Report Writer® for Windows is that reports cannot be printed in SHAMAN. The user must first exit SHAMAN and access Windows.

R&R Report Writer® for Windows documentation includes three printed manuals and a context-sensitive on-line help facility. The published references supplied with the program include:

Getting Started - Installation instructions and a quick tour of essential R&R skills.

Using R&R - Comprehensive command and procedures explanations.

Developing Applications - Using R&R reports in Windows applictions (for system developers).

All three manuals © 1993 by Concentric Data Systems, Inc.

Chapter 6

Generating Maps

Geographical Information Systems (GIS) are computer-based systems used to store and manipulate geographic information, outputting data in the form of maps or graphs.

SHAMAN uses MapInfo®, a flexible GIS, to provides the user with the ability to present a variety of geographically relevant data in the form of country, regional or district maps. This visual presentation of data is a powerful tool for analysis and synthesis.

This section provides information on outputting SHAMAN data in the form of maps, and needed references for using MapInfo®.

Producing SHAMAN Maps

SHAMAN uses MapInfo®, a flexible GIS, to provides the user with the ability to present a variety of geographically relevant data in the form of country, regional or district maps.

Before using MapInfo® you will need digitized base maps of the country you are working in, preferably in a scale of 1:2,000,000 or better, with overlays for provincial or district boundaries or their equivalent, roads, rivers, and towns and cities at various political levels.

As GIS systems become more popular, access to base maps and overlays is increasing. In Mozambique, the World Bank was the first to provide digitized maps of the country in the scale of 1:2,000,000.

An inexpensive 12" by 12" digitizing tablet is a useful tool for making overlays to go with your base maps.

To generate a map from SHAMAN data exit SHAMAN and go to Windows and MapInfo®.

Using MapInfo®

MapInfo® documentation gives complete instructions (see below) for relating SHAMAN .dbf files to base maps. As an example, suppose you wish to produce a map showing food dispatches in metric tons by District. The Partisan file is opened in MapInfo® as a .dbf file. Choose "File", "Open", then "File Format:", "dBASE DBF" and "File

Name:" "partisan.dbf". Then each dispatch record is geocoded (related) to the district to which the food was sent by choosing "Table", "Geocode" from the MapInfo® menubar. Consult the MapInfo® manuals for detailed information.

MapInfo documentation includes three printed manuals and a context-sensitive on-line help facility. The published references supplied with the program include:

Getting Started - all the information needed to start using MapInfo, including a comprehensive tutorial

MapInfo User's Guide - an overview of MapInfo's key concepts

MapInfo Reference - a detailed, alphabetic listing of MapInfo Commands

All three manuals © 1985-92 by Mapping Information Systems Corporation.

PART III

Technical Reference

Introduction

About Technical Reference

Part III of this manual contains technical information needed to install, customize, and maintain the SHAMAN system. It is divided into two chapters:

Chapter 1, Programmer Maintenance Information, includes a technical description of the SHAMAN system, and all the routines needed to intall SHAMAN and keep it running smoothly.

Chapter 2, Customizing SHAMAN, gives a suggested order and general instructions for customizing SHAMAN.

Chapter 1

Programmer Maintenance Information

Description of the SHAMAN System

Much of SHAMAN's simplicity and power comes from functions from Artful II, a relational engine that drives Clipper

SHAMAN has 10 major related DBFs and 21 indexes plus another 23 support DBFs with 27 indexes for a total of 81 open files. It is constructed with one main program source file (SHAMAN.PRG) plus additional source files (SHA_FUN.PRG, SHA_UTL.PRG, SHA_MAS.PRG, SHA_MIG.PRG, ACTTITLE.PRG AND NEEDS.PRG). Together they comprising less than 7000 lines of source code. Yet SHAMAN has the capabilities of relational find, relational query, reports, dictionary management, and system configuration, in addition to adding, editing, duplicating and deleting records, and a user-configurable help system. How is it possible? With functions from Artful II, a relational engine that drives Clipper 5. If you'd like to see the source code for SHAMAN, call up SHAMAN.PRG or the other PRG files with your favorite ASCII word processor.

SHAMAN.PRG is the main program. It sets up arrays of prompts, messages and function names, opens the dictionary, and calls DO_IT(). It also contains the data screen code for all the major table forms. These functions are called by an Artful Function THE_USUAL(), which in turn is called from the front-end menu DO_IT().

An Artful metafunction, The_Usual(), does most of the work

Many of the options in "Utilities" are library functions, and most of the file maintenance for "Support" is accomplished by a library module which can paint screens and administer SAYs and GETs for any table if you're in a hurry. Explore all the menu options. The most powerful here is the Application Dictionary - the first option in the Utilities pulldown under SYSTEM. [Be careful using it.] It is not user-bulletproof. It is not recommended that you make changes to SHAMAN's dictionary until you are a very comfortable using Clipper/Artful.

The light bar menu at the top has been created by THE_USUAL()

from the dictionary. The only entries we had to make in the DICT record for RELATIONS were filename, alias, whether to open the file, and the index key information (if we wanted to set relations, we'd have to do that too, but no relations are set here). All other fields in DICT were defaulted to calls to ArtfulLib functions.

The program code that manipulates the PLEDGE file is an example of a call to THE_USUAL(), the ArtfulLib metafunction which is doing most of the work here. There are four functions associated with the PLEDGE file:

PLE_SCRN Paints the screen background for the PLEDGE file.

PLE_SAY Displays one PLEDGE record.

PLE_KEY Edits the current or a new PLEDGE record.

PLE_FLDS Edits the non-key PLEDGE fields.

All the logic and control is from THE_USUAL() itself, and the functions it calls. You'll find the call to THE_USUAL() in SHAMAN.PRG, as a text string in the UDFs() array [THE_USUAL('PLEDGE')]. It was called by macro expansion from DO_IT(), the outer control function for SHAMAN. For example, when you select "Find", the relational find mechanism in THE_USUAL() creates, from the dictionary's relations fields, a menu of the files you can meaningfully search from this module. Once you've selected a file it constructs a menu of available index keys. Then it gets your input, and looks for the object you specified in the file you specified. "Query" is also running solely from library calls and the dictionary. You'll notice that the Query pulldown shows some selections in different colors than others, to distinguish those that can be called. "Modify", "Undo" and "Save" can only be selected after a query has been set.

Setting the Runtime Environment

For smooth operation, SHAMAN should have as much of DOS's 640K of memory available as possible

For smooth operation, SHAMAN should have as much of DOS's 640K of memory available as possible. It should also have 3 or more mb of expanded memory and many megabytes of extra hard disk space. In addition there are a number of environmental variables that must be set. Follow the directions below.

Operating System

Install DOS 6.0 as the operating system. It is more efficient and makes better use of memory than previous DOS versions.

Disk Space

Run CHKDSK /f at the dos prompt to see how much hard disk space

is free. If less than 2 or 3 megabytes are free use a hard disk manager such as XTPRO or File Manager under Windows to delete unnecessary files. Clipper 5.01's VMM (Virtual Memory Manager) works best with several megabytes of disk space available. A disk expander utility (Stacker 2.1 has been successfully used) is recommended for increasing usable hard disk space.

Three or more megabytes of extended/expanded memory will improve indexing and general system performance.

Expanded Memory

Expanded memory is quite helpful in getting the most from the system. Three or more megabytes are recommended as the indexing performance under Clipper 5.01 will be greatly improved. If the computer has only extended memory, a memory manager such as QEMM 386 by Quarterdeck can be used to convert some of the extended memory to expanded memory.

CONFIG.SYS

Ensure that enough file handles are included in the CONFIG.SYS file.

Assuming the computer used is some flavor of 80486 with 3 or more megabytes of extended memory, the following lines should be in the CONFIG.SYS file in the root directory (C:\):

```
DEVICE=C:\DOS\HIMEM.SYS
DOS=HIGH,UMB
DEVICEHIGH=C:\DOS\EMM386.EXE 2048 RAM
DEVICEHIGH=C:\DOS\RAMDRIVE.SYS 1024/E
SHELL=C:\DOS\COMMAND.COM C:\DOS\ /E:2048 /P
FILES=171
BUFFERS=30
```

The most important thing here is that the Files=171 line be included in the Config.Sys file. In DOS 6.x, MakeMem can be run to optimize use of RAM. Disk-caching should also be used to increase the performance of SHAMAN.

AUTOEXEC.BAT

Again assuming an 80486, the following lines should be included in the AUTOEXEC.BAT file in the root directory.

```
C:\NG\EH -U -O
SET CLIPPER=F171
```

System Installation

Ensure that you have installed the minimal hardware configuration and followed the above instructions for Setting the Runtime Environment. You are now ready to install SHAMAN.

The initial SHAMAN installation is quite easy. Insert the SHAMAN program disk in the A: Drive and type "A: <ENTER>" followed by "INSTALL <ENTER>". An executable version of PKUNZIP will

automatically unzip all the compressed SHAMAN files, make necessary sub-directories off the c: drive to put them into and attempt to start SHAMAN.

Initial SHAMAN installation is quite easy

If the CONFIG.SYS file has been previously modified (and the computer restarted) to have the line FILES=171, SHAMAN will present the opening user name and password dialog box. Typing "SHAMAN" for the user name and "shaman" for the password will allow initial entry into the system. SHAMAN will automatically construct index files for all the files it opens. This index file construction is usually only done once when the index files are initially absent. This is also one method of reindexing the whole system if problems are encountered with running SHAMAN. Simply delete all the files with a .NTX extension and reopen SHAMAN.

For full access to all of SHAMAN's modules, enter the system with the user name "DAVID" and password "may". This allows the system administrator to enter the System Administration module in the "Utility", "System" section and then change the passwords and user names to suit the current installation. The system administrator should make him/herself *user number 1* with a secure password. The user named "SHAMAN" is *user number 3* and has the user level of which allows read-only access to most modules.

Experiment with existing SHAMAN data files to become familiar with the system.

After the files have been successfully installed, the system administrator should experiment with navigating SHAMAN and performing data entry in various modules until he/she is comfortable with using SHAMAN's many function. Then the system needs to be customized for the country or location using the system. Instructions on customization may be found in Part I, Chapter 3 and Part II, Chapter 4.

The basic SHAMAN system is supplied in the form of an executable

System Maintenance

file SHAMAN.EXE. It is already compiled and linked. The following instructions, therefore are only necessary in the case of a program modification where SHAMAN.PRG or the other *.PRG have been changed.

How to Compile the SHAMAN Program

The current version of SHAMAN was compiled with Clipper® 5.01 using Artful II® version 2.7 and linked with Blinker® 2.1. Before any compiling or linking of SHAMAN can be done, Clipper 5.01, Blinker 2.1 or greater and Artful II version 2.7 must be installed following manufacturers recommended installation procedures.

To make a modified version of SHAMAN, he following programs need to be compiled: SHAMAN.PRG, SHA_FUN.PRG,

SHA_UTL.PRG, SHA_MAS.PRG, SHA_MIG.PRG, ACTTITLE.PRG andNEEDS.PRG. Compile them with the following series of commands from DOS while in the subdirectory C:\CLIPPER5\ARTFUL\SHAMAN:

First ensure that the following files are in the subdirectory C:\CLIPPER5\ARTFUL\SHAMAN:

```
DZ_SINON.CH
DZ_CONFG.CH
DZ_LOGON.CH
AA_DUST1.CH
```

These are header files needed by the preprocessor to compile SHAMAN files.

Then type the following commands:

```
SETAA
SETAPP SHAMAN
CLIPPER SHAMAN /n/l/w
CLIPPER SHA_FUN /n/l/w
CLIPPER SHA_UTL /n/l/w
CLIPPER SHA_MAS /n/l/w
CLIPPER SHA_MIG /n/l/w
CLIPPER ACTITLE /n/l/w
CLIPPER NEEDS /n/l/w
```

SHAMAN.OBJ, SHA_FUN.OBJ, SHA_UTL.OBJ, SHA_MAS.OBJ, SHA_MIG.OBJ, ACTTITLE.OBJ, and NEEDS.OBJ should have been successfully created by the compiler.

How to Link the SHAMAN Programs

SHAMAN was originally linked using Blinker 2.0. It could be linked with Clipper's RTLINK or with another compatible linker, but for best results (no surprises) Blinker is recommended.The following object files must be in the C:\CLIPPER5\ARTFUL\SHAMAN subdirectory:

```
SHAMAN.OBJ
SHA_FUN.OBJ
SHA_UTL.OBJ
SHA_MAS.OBJ
SHA_MIG.OBJ
ACTTITLE.OBJ
NEEDS.OBJ
```

In addition SHAMANB.LNK must be in the same subdirectory. It is an ASCII file with the following lines:

```
blinker incremental off
output shaman.exe
fi SHAMAN
```

```
fi sha_fun
fi sha_utl
fi sha_mig
fi sha_mas
fi needs
fi acttitle
lib clipper
lib shamanb
lib artful
lib artfulc
lib overcl
lib pckernel
lib pcfuncs
```

The lines that start with "f i" list all the object files that will be linked while the "l i b" lines tell the linker what libraries to look in for the compiled functions needed by the program object code modules. The order specified is important. The library modules with extensions of LIB are found in the subdirectory C:\CLIPPER5\ARTFUL\LIB or C:\CLIPPER5\LIB.

The actual linking is accomplished with the following command:

```
BLINKER @SHAMANB
```

RTLINK is the linker that came with Clipper 5.01 and is located in the subdirectory C:\CLIPPER5\BIN. Blinker should be in the same subdirectory.

How to make a program disk

The program PKZIP.EXE and ZIP2EXE.EXE should be installed in a subdirectory named C:\ZIP. An updated program disk can be easily made by going to the subdirectory C:\CLIPPER5\ARTFUL\SHAMAN\PROG. Type MAKEPROG <ENTER>. A self-expanding file containing all the compressed SHAMAN files will be made. It is named SHA_PROG.EXE. Copy this file and the INSTALL.BAT file to a new floppy and you now have the whole SHAMAN system, including data, on a single diskette ready to install on another computer or to use to update a previously-installed SHAMAN system.

DPD (Designer PullDown MenuMaker)

DPD (Designer PullDown MenuMaker) is the Artful II program used to construct the SHAMAN main menu. It can be used to modify the behavior of the system. The files involved are SHAM_MNU.DBF, SHAM_MNU.DBT and SHAM_MN1.NTX. Directions for the use of Artful II's DPD are found in the Artful II manual.

WinGen (Windows Screen Designer and Generator)

WinGen (Window Generator) is the Artful II program used to construct the NEEDS and ACTTITLE SHAMAN full-screen maintenance screens. It allows the user to design all aspects of a modules screen and then generate the code necessary to run the module. Directions for the use of Artful II's WinGen are found in the Artful II manual.

System Integrity

Initial setup and ongoing maintenance of SHAMAN's support files is important for system integrity.

SHAMAN's support files, combined with picklists which act as authority tables for specific fields, provide verification for much of the data entry. Initial setup and ongoing maintenance of these support files is important for system integrity. Additions may have to be made from time to time. For example, a new donor or a new location may have to be added. The system administrator needs to review these authority tables regularly to ensure that no alternate spellings or acronyms have been added.

In addition, data entry verification procedures need to be set up and followed rigorously. It is a good idea to have one person checking another's data entry by checking a printout of the data against the data collection forms.

A daily log of each data entry clerk's activities should be kept, logging in what files were modified by adding or editing. SHAMAN's audit file keeps a record of every change made in SHAMAN.

Data entry and verification procedures must be set up and followed rigorously. Text editing is also very important.

Data entry is likely to be done by national staff having English as a second language. This means that the extensive text in the various memo fields needs to be edited by a native English speaker. This is especially important in the CHAP, as the important descriptive text fields become part of widely-disseminated documents. However, language errors cause any report to lose force.

Troubleshooting

File Handles

Often when things don't work, or SHAMAN is unable to index its files when it first starts up, the system does not have enough file handles. Make sure there is a line in the CONFIG.SYS file that says "FILES=171".

Memory Management

If SHAMAN ceases to work for no apparent reason, or you get an error message that says something about memory, it may mean that someone has added a new TSR program that is using up valuable DOS

memory. Ensure with memory management software that there are at least 570K of usable RAM available for the system.

Use of SHARE in a Network Environment

If the DOS program, SHARE is loaded, make sure that the /L parameter is set up to the number of file handles, e.g.,

```
SHARE /F:20000 /L:171
```

See the DOS manual for a more detailed explanation of SHARE and its use.

Index File Corruption

If you suddenly start finding that you can't find records that you know are in the system, or other strange related behavior occurs, it probably means that some indexes have been corrupted.

The Utilities menu contains a reindex function in which individual indexes can be reindexed

OR: you can cause indexing of all SHAMAN files by deleting all of them from the DOS prompt with the command `DEL *.NTX`.

Memo File (.DBT) Bloat

Memo file bloat will most often occur with the ACTIVITY.DBT file. If you notice the file size increasing to much more than 500K, it means that extensive editing of activity memo fields has caused undue file bloat. This can be corrected by by using the Clipper utility DBU:

1. Call up the Activity file in DBU.
2. Copy the Activity file to another file of any temporary name, e.g., ACT_TEMP.
3. Ensure that the new file has all the Activity file information in it.
4. Delete the original ACTIVITY.DBT file.
5. Rename the temporary ACT_TEMP.DBT file ACTIVITY.DBT.

The ACTIVITY.DBT file should now be considerably smaller than it was.

Useful Addresses

ARTFUL II/GOLD	Artful Applications Inc. 176 George St. Toronto, Ont. M5R 2M7 Canada Phone: (416) 920-7395 FAX: (416) 920-4892
BLINKER	Blink Inc. 8001 W. Broad St. Richmond VA 23294 USA Phone: (804) 747-6700 FAX: (804) 747-4200
CLIPPER	Computer Associates Inc. One Computer Associates Plaza Islandia, NY 11788-7000 USA Phone: 1-800-225-5224
MAPINFO	MapInfo Corporation 200 Broadway Troy, NY 12180 USA Phone: (518) 274-6000 FAX: (518) 274-6000
PK ZIP/PK UNZIP	PKWARE Inc. 9025 N. Deerwood Dr. Brown Deer, WI 53223 USA Phone: (414) 354-8699 FAX: (414) 354-8559
R&R REPORT WRITER	Concentric Data Systems Inc. 110 Turnpike Rd. Westborough, MA 01581 USA Phone: (508) 366-1122 FAX: (508) 366-2954

Chapter 2

Customizing SHAMAN

SHAMAN can be customized to meet country-specific and agency/program-specific needs. See Part I, Chapter 3, for general information on the analysis phase of customizing SHAMAN. See Part II, Chapter 4 for instructions on using the support modules needed to customize SHAMAN.

This chapter gives general instructions for customizing the various files. By following the suggested order, you should have SHAMAN up and running quickly.

Defaults File

The file as received already has data in all the default fields. Most of these defaults are headers which appear as field names on SHAMAN screens. For example, "District" and "Province" are headers that appear in many screens. They reflect subdivisions used in many but not all countries. You may need to change them to "County" and "State", or some other political subdivision.

The Defaults table, accessed under the Utility menu is organized in order of the categories:

General Defaults
Population File Defaults
Partisan File Defaults
Pledge File defaults
Mailing List defaults

The Defaults file fields for each of these categories are listed below along with example data from Mozambique and a general description:

General Defaults

Field	Contents	Description
Project_No	Project No.	Project No. (if used)
RR_Hdr	UNOHAC SHAMAN	R&R Reports - Header
Sys_Name	SHAMAN:System for Humanitarian Assistance Management	SystemName
Office	MAPUTO	Office Default
Office_Hdr	UNOHAC Office	Office Header (used in R&R reports)
Office_Dir	Bernt Bernander	Office Director Name
Country	MOZAMBIQUE	Country where SHAMAN in use
Humanas_FY	93/94	Humanitarian Aid Fiscal Year
District	District	District level Header
Province	Province	Province level Header

Note: These general defaults appear in many files.

Population File Defaults

Field	Contents	Description
Age_1_Hdr	GOM Plan/Demog Dec/92:	Headers for agencies whose population figures are being used.
Age_2_Hdr	GOM Dec/95 Projection:	
Age_3_Hdr	MSF:	
Age_4_Hdr	'92 Tot Beneficiaries:	
Age_5_Hdr	Agency 5 Population:	
Age_6_Hdr	Agency 6 Population:	
Pop_1_Hdr	1980 Census Poulation	
Pop_2_Hdr	Oct 1992 Population	

Note: See the Population File module screen to understand these defaults.

Partisan File Defaults

Field	Contents	Description
PartisanHd	RENAMO	Name used by partisan group.

Note: See the Partisan File module screen to understand these defaults.

Pledge File Defaults

Field	Contents	Description
OtherLang	Portuguese	Language used for other-language summary.
CarryOv_Hd	92 93 Carryover	Last year carryover.
Cur_Com_Hd	93 94 Commitment	Current year commitment.
Nex_Com_Hd	94 95 Commitment	Next year commitment.
Disburs_Hd	Oblig.To Disburse	Obligated $ amount.
CHAP_FY	93 94	Fiscal Year.

Note: See the CHAP module screen to understand these defaults

Mailing List Defaults

Field	Contents	Description
Mail_List1	Bull	Names of publications or documents being distributed or mailed.
Mail_List2	CHAP	
Mail_List3	LIST 3	
Mail_List 4	LIST 4	
Mail_List5	LIST 5	
Mail_List6	LIST 6	
Mail_List7	LIST 7	
Mail_List8	LIST 8	

Note: See the Mail File module screen to understand these defaults.

The specifications for each of these fields (how many characters are allowed) can be found in the data dictionary.

The files listed below should have data entered in the suggested order. Instructions and sample screens can be found in Part II, Chapter 4.

Activity Title File

Information for this mission critical table is decided during the analysis phase. It includes the Activity Number, and long and short titles.

Note: The Activity Title file cannot be modified after setup without serious consequences for the integrity of SHAMAN. It is possible to add a new activity to a sector, or even a new sector to a category, as long as they are not inserted into a run of existing numbers. However, changing numbers or titles, or even shifting them will affect every related file and report.

Needs File

The Needs file is closely related to the Activity Title file. Each Activity Title record has one or more child Needs records. A Needs record duplicates the Activity Number of its parent record, and contains a specific Needs title and the needed amount in US$.

Setting up the Needs file is an important part of initial SHAMAN customization. New records can be added as required.

Note The Needs file is one of the files that would be seriously affected by any change made in the Activity Title file. All Needs records would have to be altered accordingly

Activity File

The Activity file is the main database file of SHAMAN. It contains one record for every humanitarian assistance program activity. It must be customized with the data and text decided upon during the analysis phase.

The SHAMAN sytem is dependent on this database file (no pledge records can be entered without a related activity record). To initialize the system, the Activity file records are entered under the Utility -> Support Files ->Activity File menu choice.

Donor

Many of the donors in the list supplied will remain the same. Add or delete as required.

Agency

Many of the agencies in the list supplied will remain the same. Add delete as required.

Location Type

Some location types may remain the same, e.g., "DS, District Seat." Others will need to be added or deleted.

Location

Begin by entering all the main cities, towns, and villages of known humanitarian assistance activities, ensuring that you choose a standard spelling. Other locations can be added as needed, in some cases by zooming from the data-entry screen.

Note. Because locations can be added from data-entry screens, it is important to review this authority file regularly to prevent duplication through non-standard spelling

Resource Type

The resource sectors and types to be tracked are determined during the analysis phase.

Incident Type

The standard list in the file should be sufficient.

Mine Type

The standard list in the file should be sufficient to start. Others can be entered by zooming from the data entry screen.

Mode of Transport

The standard list in the file should be sufficient.

PART IV

Appendices

Appendix A

Program Code

SHAMAN.PRG
SHA_MAS.PRG
SHA_MIG.PRG
ACTTITLE.PRG
NEEDS.PRG
SHA_FUN.PRG
SHA_UTL.PRG

```
*: Procedure file: C:\CLIPPER5\ARTFUL\SHAMAN\SHAMAN.PRG
*:         System: SHAMAN: System for Humanitarian Assistance Management
*:
*:*************************************************************
*
*
*  ╔═══════════════╦═══════════════════════════════════════════╗
*  ║ Program       ║ SHAMAN.PRG                                ║
*  ║ System        ║ ARTFUL.LIB for Clipper 5.01               ║
*  ║ Purpose       ║ Integrated UNOHAC Database Management System ║
*  ║ Conventions   ║ Uses  THE_USUAL() and dictionary development ║
*  ║ Syntax        ║ SHAMAN [/mono]                            ║
*  ║ Parameters    ║                                           ║
*  ║    mono       ║ pass to force monochrome setting          ║
*  ║ Artful Calls  ║                                           ║
*  ║ Author        ║ David W. Zimmerly                         ║
*  ║ Copyright     ║ BlueWater Software, 1993                  ║
*  ╠═══════════════╩═══════════════════════════════════════════╣
*  ║ Notes:                                                    ║
*  ╚═══════════════════════════════════════════════════════════╝
*
#include "Box.ch"
#include "Artful.ch"
#include "AA_Pulld.ch"
#include "AA_Gets.ch"
#include "FileIO.ch"

// References to data arrays from UGet():
#xtranslate M( <foo> ) => aRecord\[ FieldPos( <"foo"> ) \]
#xtranslate aMenuStruct() => ;          // dbf structure of a menu file
       { ;
      { "MENU_ID",    "C", 15, 0 }, ;
      { "PROMPT",     "C", 40, 0 }, ;
      { "CHILD_ID",   "C", 15, 0 }, ;
      { "BLOCK",      "C", 80, 0 }, ;
      { "MESSAGE",    "C", 80, 0 }, ;
      { "HELP_TEXT",  "M", 10, 0 }, ;
      { "SPECS",      "M", 10, 0 }, ;
      { "SECURITY",   "M", 10, 0 }, ;
      { "EXECUTE",    "L",  1, 0 }, ;
      { "HOTKEY",     "N",  3, 0 }, ;
      { "SELECTABLE","C", 80, 0 }, ;
      { "MONOCHROME","C", 40, 0 }, ;
      { "COLOR",      "C", 40, 0 }, ;
       }

#define NHANDLES  171
#define CAPS      '@K!'
#define EXE_NAME  "S H A M A N"
#define EXE_DESC  "System for Humanitarian Aid Management"
#define AACLR_BOX "GR+/B"

// Statics
Static  aMenuStru := aMenuStruct()
Static  aEntire   := {}           // keep track of whole menu here
Static  aCols     := {}           // array of main menu columns
Static  aImages   := {}           // array of menu images
Static  isHelpSeen:= .F.          // show help?
Static  aCurr     := {}           // current submenu
Static  nCurr     := 0            // pointer into aImages
Static  IsSaved   := .T.          // menu altered since last save?
Static  nSpacing  := 1            // adjustable spacing! wot a concept
Static  nMenuNo   := 1            // main menu item selected
Static  IsHelpMsg := .T.          // display F1 = Help if there's room
Static  cCurMenu  := ""           // name of last menu file loaded
Static  IsTopMenu := .T.          // are we on the top menu?
```

```
Static  nStateNo  := 0              // idle state

Static  DictFile  := 'SHA_DICT'          // dictionary filename
Static  mUserNum
Static  mPermission

// Field declarations for MENUS.MNU
FIELD Menu_id, Prompt, Child_id, Block, ,
      Message, Help_Text, Specs, Security, Execute ;
      IN MENUS



* ============================================================
Function Main
* ============================================================
MemVar mUserNum, mPermission
MemVar F3_Toggle, F7_Toggle, Dict

Local aMenu    := BuildMenu( "MAIN", "SHAMAN.MNU" )
Local cColor1  := "N/GB,GB+/N,B,B,GB/N"
Local cColor2  := "GB+/W,GR+/N,B,B,GB/N"
Local cOld_Help := HelpCode( ProcName() ), nOldOrder

Local Logo_Line := "BlueWater Software, " + ;
           Strint(Year(Date() )) + " - Ver 0.99" // Manecas included:
                                  // UPDNLOAD() 16/09/93
Local Monochrome

   Monochrome := ( pcount() > 0 )
   ArgSet( ( Monochrome ) )

   Set SoftSeek Off
   Set ScoreBoard Off
   Set Deleted On
   Set Exact Off
   Set Date British
   SetBlink(.F.)

   Public F7_Toggle := .F.               // F7 for color chart in SYS

   Setup(NHANDLES)
   Clear Screen
   SysColor(AACLR_NORMAL)

   DeskTop()                                    // startup screen

   If ! LogIn("SHAMAN",, "SHA_SYS")
      Quit
   EndIf
   nOldOrder:= SetOrder()

   Public Dict
   Dict := "SHA_DICT"
   mPermission := Sys->Permission
   mUserNum    := Sys->UserNum

   Set Key K_ALT_A To Tag_Record         // Menu to Tag or Untag all
   Set Key K_ALT_T To Tag_Record         // Tag a record
   SET KEY K_ALT_U TO TAG_RECORD                // Untag a record
   SET KEY K_ALT_X TO Imm_Exit           // Exit from program

   FontInstall(.t., .t.)
   Sign_On( EXE_NAME,EXE_DESC, Logo_Line )

   If !Chk_Dict( DictFile )                    // open the dictionary
```

```
      Set Cursor On
      Quit
   EndIf


   SysColor( 1 )
   SetBlink( .F. )

   Do While .T.
      Set Wrap On
//       SysColor()
      DeskTop()
      IF !Empty( aMenu )              // so this is not permissible
      PdActivate( aMenu )
      Endif
      If Verify( "Leave SHAMAN and return to DOS" )
        Exit
      EndIf
   EndDo

   Sign_Off( EXE_NAME,EXE_DESC, Logo_Line )

   HelpCode( cOld_Help )
   SetOrder ( nOldOrder )

Return Nil


* ============================================================
Function Imm_Exit                  // ALT-X immediate exit from program
* ============================================================
   Sign_Off( EXE_NAME,EXE_DESC )
   QuitProg()
Return .T.


* ============================================================
Function Par_Init                  // Partisan Initialization
* ============================================================
   The_Usual( 'Partisan' )
Return .T.


* ============================================================
Function Mine_Init                  // Mine Assessment Initialization
* ============================================================

   SetBlink( .F. )
   The_Usual( 'Mine_Ass' )

Return .T.


* ============================================================
Function Sys_Init                              // optional initializations
* ============================================================
MemVar mUserNum

   If mUserNum == "1    " .or. mUserNum == "2    "
   Else
      Pop_Msg("Contact System Operator for Help")
      Return .F.
   EndIf
   The_Usual('Sys')

Return .T
```

```
*
Function Dct_Init                                   // optional initializations
*
MemVar mUserNum

   If mUserNum == "1   " .or. mUserNum == "2   "
   Else
      Pop_Msg("Contact System Operator for Help")
      Return .F.
   EndIf
   Dct_Maint( "sha_dict" )

Return .T.


*
Function ActTitle_Init                              // optional initializations
*
MemVar mUserNum

   If mUserNum == "1   " .or. mUserNum == "2   "
   Else
      Pop_Msg("Contact System Operator for Help")
      Return .F.
   EndIf
   The_Usual( "ActTitle" )

Return .T.


*
Function Needs_Init                                 // optional initializations
*
MemVar mUserNum

   If mUserNum == "1   " .or. mUserNum == "2   "
   Else
      Pop_Msg("Contact System Operator for Help")
      Return .F.
   EndIf
   The_Usual( "Needs" )

Return .T.


*
Function Demob_Init                    // Demobilization Initialization
*
   Do_Swap('\clipper5\artful\demob\demob', 512 ,, '',.T. , DictFile)
   SetBlink( .F. )
   FontInstall(.t., .t.)
Return .T.


*
Function RR_Init                       // R&R Initialization
*
   Do_Swap('\RR\RR work1/C',512 ,, '',.T. , DictFile)
   SetBlink( .F. )
   FontInstall(.t., .t.)
Return .T.


*
```

```
Function ViewCredits                                    // optional initializations
* ═══════════════════════════════════════════════════════════════════════
Local cTempScrn := SaveScreen( 4, 4, 22, 77 )
Local cSaveColor := SetColor()
   Boxer( 4, 4, 21, 76, 2,, " ", "BG/N",, .T., .T. )
   SysColor( "BG/N" )
   @ 6,32 Say "BlueWater Software"
   @ 7,31 Say "12 Shadetree Crescent"
   @ 8,30 Say "Nepean, Ontario K2R 7E2"
   @ 9,29 Say "CANADA Tel: (613) 228-1268"
   @11,33 Say "Acknowledgements"
   @12,33 Say "────────────────"
   @14, 6 Say "Systems Analysis, Design and Development.....David W. Zimmerly, Ph.D."
   @15, 6 Say "Help Screens and SHAMAN Manual..............Helga G. Zimmerly"
   @17, 6 Say "UNHAC Staff in Maputo assisted in the development and testing phases"

   SysColor( AACLR_HEADER )
   @ 4,33 Clear To 4,47
   @ 4,34 Say "C R E D I T S"

   SysColor( "W/N" )
   @21,27 Clear To 21,53
   @21,28 Say "Press any key to continue"

   Inkey( 0 )   // wait until a key is pressed
   RestScreen( 4, 4, 22, 77, cTempScrn )
   SetColor( cSaveColor )

Return .T.


* ═══════════════════════════════════════════════════════════════════════
Function DOSGate
* ═══════════════════════════════════════════════════════════════════════
Local cTempScrn := SaveScreen( 0, 0, MaxRow(), MaxCol() )
Local lResult := .T.

   SysColor( "W+/B" )
   Cls
   Do_Swap('', 0,, '')
   RestScreen(0, 0, MaxRow(), MaxCol() , cTempScrn )

Return lResult


* ═══════════════════════════════════════════════════════════════════════
Function Add_Scr
* ═══════════════════════════════════════════════════════════════════════
Local SubHeader := "W+/B"

   SysColor( AACLR_BOX )
   Boxer( 3, 0, 23, 79, 2,,, "GR+/B" ,, .T. )

   SysColor( AACLR_NORMAL )
   @ 5, 3 Say "Company:                                          Classification:"

   SysColor( AACLR_HEADER )
   @ 3,33 Clear To 3,47
   @ 3,34 Say "A D D R E S S"

   SysColor( AACLR_NORMAL )
   Boxer( 7, 3, 10, 29, 1,,, "W+/B" ,, .T. )
   SysColor( SubHeader )
   @ 7, 5 Say " NAME "

   SysColor( AACLR_NORMAL )
```

```
   @ 8, 6 Say "Last:"
   @ 9, 6 Say "First:"

   Boxer( 8, 55, 14, 77, 1,,, "W+/B" ,, .T. )
   SysColor( SubHeader )
   @ 8,57 Say " TELEPHONE "

   Boxer( 11, 3, 17, 53, 1,,, "W+/B" ,, .T. )
   SysColor( SubHeader )
   @11, 5 Say " Address "

   SysColor( AACLR_NORMAL )
   @12, 6 Say "Address 1:"
   @13, 6 Say "Address 2:"
   @14, 6 Say "City:"
   @15, 5 Say Replicate( " ", 10 - Len(Trim(Defaults->Province))) + ;
          Proper(Trim(Defaults->Province))+":"
   @16, 6 Say "Country:"

   Boxer( 15, 55, 18, 77, 1,,, "W+/B" ,, .T. )
   SysColor( SubHeader )
   @15,57 Say " TELEX "

   Boxer( 18, 3, 22, 39, 1,,, "W+/B" ,, .T. )
   SysColor( SubHeader )
   @18, 5 Say " CONTACT "

   SysColor( AACLR_NORMAL )
   @19, 6 Say "Name:"
   @20, 6 Say "Title:"
   @21, 6 Say "Phone:"

   Boxer( 19, 55, 22, 77, 1,,, "W+/B" ,, .T. )
   SysColor( SubHeader )
   @19,57 Say " FAX "

Return .T.

* ============================================================
Function Add_Key
* ============================================================
MemVar mPermission
Local lResult    := .F.
Local GetList    := {}


   SysColor(2)     // Gets

   If mPermission <> "EXPERT"
      Return .F.
   EndIf

   @ 5,12 Get mCompany Picture CAPS
   @ 8,13 Get mLast_Name Picture CAPS
   @ 9,13 Get mFirst_Name Picture CAPS

   SysColor(1)          // Normal

Return Read_It( ,,GetList)


* ============================================================
Function Add_Fld
* ============================================================
MemVar mPermissiob, mClassify
```

```
Local lResult     := .F.
Local Flds_4      := { "ALIAS", "Proper(Classify)" }, ; // Classification array
      Cla_Header  := { "Code", "Classification" }
Local GetList     := {}

   SysColor(2)     // Gets

   If M->Is_Append
      Aud_Before()
   EndIf

   If mPermission <> "EXPERT"
      Return .F.
   EndIf


//  A = Agency
//  D = Donor
//  E = Employee
//  M = Media
//  O = Other
//  S = Supplier
//  T = Transporter

   @ 5,66 Get mClassify BOX B_SINGLE;
              Picture CAPS ;
              Valid Help_Fill( @mClassify, "Classify", Flds_4 ;
                       ,,,,,,, Cla_Header,,,1 ,1 )

   @12,17 Get mAddress1 Picture CAPS
   @13,17 Get mAddress2 Picture CAPS
   @14,17 Get mCity Picture CAPS
   @15,17 Get mProvince Picture CAPS
   @16,17 Get mCountry Picture CAPS

   @19,13 Get mContact Picture CAPS
   @20,13 Get mCon_Title Picture CAPS
   @21,13 Get mCon_Phone

   @ 9,57 Get mPhone_1
   @10,57 Get mPhone_2
   @11,57 Get mPhone_3
   @12,57 Get mPhone_4
   @13,57 Get mPhone_5

   @20,57 Get mFax_1
   @21,57 Get mFax_2

   @16,57 Get mTelex_1 Picture CAPS
   @17,57 Get mTelex_2 Picture CAPS

   lResult := Read_It( ,,GetList )

   If lResult
      If M->Is_Append
      Else
      If UpDated()
        Aud_After()         // Check Audit Trail for changes
      EndIf
      EndIf
   EndIf

Return lResult
```

```
Function Add_Say
* ═══════════════════════════════════════════════════════════════
Local lMemo     := F.
Local GetList := {}

   SysColor( 2 )                                    // hilighted video
   * Set Keys for tagging and untagging
   Set Key K_ALT_A To Tag_Record          // Menu to Tag or Untag all
   Set Key K_ALT_T To Tag_Record          // Tag a record
   SET KEY K_ALT_U TO TAG_RECORD                 // Untag a record
   SET KEY K_ALT_X TO Imm_Exit            // Exit from program

   SYSCOLOR ( "N/W" )
   @ 5,12 Say Proper( Address->Company )
   @ 6,13 Say Proper( Address->Last_Name )
   @ 9,13 Say Proper( Address->First_Name )

   @ 5,66 Say Fetch( Address->Classify, "Classify", "Classify" )

   @12,17 Say Proper( Address->Address1 )
   @13,17 Say Proper( Address->Address2 )
   @14,17 Say Proper( Address->City )
   @15,17 Say Proper( Address->Province )
   @16,17 Say Proper( Address->Country )

   @ 9,57 Say Address->Phone_1
   @10,57 Say Address->Phone_2
   @11,57 Say Address->Phone_3
   @12,57 Say Address->Phone_4
   @13,57 Say Address->Phone_5

   @16,57 Say Address->Telex_1
   @17,57 Say Address->Telex_2

   @20,57 Say Address->Fax_1
   @21,57 Say Address->Fax_2

   @19,13 Say Proper( Address->Contact )
   @20,13 Say Proper( Address->Con_Title )
   @21,13 Say Address->Con_Phone

   * Determine If memo field has an entry
   lMemo := MemoStatus(Address->Remarks)
   @ 04, 73 CheckBox lMemo

Return .T.


* ═══════════════════════════════════════════════════════════════
Function Par_Scr
* ═══════════════════════════════════════════════════════════════
Local nHdrLen := Len( Expand(Trim( Defaults->PartisanHd ) ) )

   SysColor( AACLR_BOX )
   Boxer( 7, 7, 20, 72, 2,,, "GR+/B" ,, .T. )
   @12, 7 Say "╟"
   @12,72 Say "╢"
   @15, 7 Say "╟"
   @15,72 Say "╢"
   @12, 8 Say "──────────────────────────────────────────────────────────────"
   @15, 8 Say "──────────────────────────────────────────────────────────────"

   SysColor( AACLR_NORMAL )
   @ 9,10 Say "Location:                              Agency:"
   @10,10 Say Replicate(" ",8-Len(Trim(SubStr(Defaults->District,1,8)))) + ;
              Proper(Trim(SubStr(Defaults->District, 1, 8))) + " "
```

```
   @11,10 Say Replicate(" ",8-Len(Trim(SubStr(Defaults->Province,1,8)))) + ;
            Proper(Trim(SubStr(Defaults->Province, 1, 8))) + ":"
   @11,61 Say "Memo:"
   @13,10 Say "Dispatches:                To           (dd/mm/yy)"
   @14,10 Say "Transported by:"
   @16,10 Say "Commodity Type:"
   @17,10 Say "Item(s) "
   @18,10 Say "Quantity:"
   @19,10 Say "Measurement:"

   SysColor( AACLR_HEADER )
   @ 7,(40 - nHdrLen / 2) Clear To  7, (40 + nHdrLen / 2) + 1
   @ 7,(40 - nHdrLen / 2) + 1 Say Expand( Defaults->PartisanHd )

   SysColor( AACLR_NORMAL )

Return .T.


* ============================================================
Function Par_Key
* ============================================================
MemVar mPermission, mTag

Local lResult    := .F.
Local GetList    := {}


   SysColor(2)     // Gets

   If mPermission <> "EXPERT"
      Return .F.
   EndIf

   mTag := .T.             // TAG true If edit or add

   If M->Is_Append
   Else
   EndIF

   SysColor(1)          // Normal

Return Read_It( ,,GetList)

* ============================================================
Function Par_Fld
* ============================================================
MemVar mProvince, mDistrict, mLocation, mAgency, mCategory, mUnit
MemVar mPermission, mMode, mQuantity

Local lResult    := .F.
Local GetList    := {}
Local Flds_2     := { "LOCATION", "District", "Province" }, ; // Location array
      Loc_Header := { "Location",  Proper(Trim(Defaults->District)), ;
                      Proper(Trim(Defaults->Province)) }, ; // Location header
      Flds_3     := { "AGENCY", "Proper(AgencyLong)" }, ;     // Agency array
      Age_Header := { "Agency;(Short Name)", "Agency;(Long Name)" }, ;
      Flds_4     := { "ALIAS", "Proper(Mode)" }, ;     // Mode array
      Mod_Header := { "Mode;Code", "Mode of;Transport" }
Local LocFilter  := "Location->Location != ' ' "
Local aCatList   := { "FOOD ", ;
                      "SEEDS", ;
                      "R&S  ", ;
                      "TOOLS"  ;
                    }
Local aUnList    := { "MT   ", ;
```

```
            "UNITS"  ;
           )

   SysColor(2)      // Gets

   If M->Is_Append
      mProvince := Space( 20 )
      mDistrict := Space( 20 )
      mCategory := "FOOD "
      mMode     := "Truck"
      mUnit     := "MT   "
   Else
      Aud_Before()
   Endif

   If mPermission <> "EXPERT"
      Return .F.
   EndIf

   @ 9,20 Get mLOCATION BOX B_SINGLE;
              Picture CAPS ;
              Valid Help_Fill( @mLocation, "Location", Flds_2 ;
                       ,, LocFilter,,,,, Loc_Header, ;
                       ( || GetOrder("LOCATION") ) ;
                       ,, 1, 1 )  and. ;
            Pro_Loc( mLocation, @mDistrict, 10, 20, @mProvince, 11, 20 )

   @ 9,50 Get mAgency BOX B_SINGLE;
                    Picture CAPS ;
              Valid Help_Fill( @mAgency, "Agency", Flds_3 ;
                       ,,,,,,, Age_Header,1,, 1, 1 )  .and. ;
            Pro_Loc( mLocation, @mDistrict, 10, 20, @mProvince, 11, 20 )


// Province and District are automatically inserted by the Pro_Loc Function.
// @11,20 Get mProvince Picture CAPS
// @10,20 Get mDistrict Picture CAPS

   @13,26 Get mFrom Picture "@D"
   @13,40 Get mTo Picture "@D"

   @14,26 Get mMode BOX B_SINGLE;
                    Picture CAPS ;
              Valid Help_Fill( @mMode, "Mode", Flds_4 ;
                       ,,,,,,, Mod_Header,1,, 1, 1 )


   @16,26 Get mCategory DropField aCatList ;
                     Valid !Empty( mCategory )

   @17,26 Get mItem Picture CAPS

   @18,26 Get mQuantity Picture [99999999.999] ;
                     Valid mQuantity > 0

   @19,26 Get mUnit DropField aUnList ;
             Valid Unit_Sub( @mUnit )

   lResult := Read_It( ,,GetList )

   If lResult
      If M->Is_Append
      Else
      If UpDated()
        Aud_After()          // Check Audit Trail for changes
      EndIf
```

```
      EndIf
   EndIf

Return lResult


* ============================================================
Function Unit_Sub( cUnit )
* ============================================================
Local lResult := .T.
   cUnit := Substr(cUnit,1,2)

Return lResult


* ============================================================
Function Par_Say
* ============================================================
Local lMemo    := .F.
Local GetList := {}

   SysColor( 2 )                                  // hilighted video
   * Set Keys for tagging and untagging
   Set Key K_ALT_A To Tag_Record          // Menu to Tag or Untag all
   Set Key K_ALT_T To Tag_Record          // Tag a record
   Set Key K_ALT_U To Tag_Record                 // Untag a record
   Set Key K_ALT_X To Imm_Exit            // Exit from program

   SYSCOLOR ( "N/W" )
   @ 9,20 Say Proper( Partisan->Location )
   @ 9,50 Say Partisan->Agency
   @10,20 Say Proper( Fetch( Partisan->Location, "Location", "District", 1 ) )
   @11,20 Say Proper( Fetch( Partisan->Location, "Location", "Province", 1 ) )
   @13,26 Say Partisan->From
   @13,40 Say Partisan->To
   @14,26 Say Proper(Fetch(SubStr(Partisan->Mode,1,1), "Mode", "Mode", 1 ))
   @16,26 Say Partisan->Category
   @17,26 Say Proper( Partisan->Item )
   @18,26 Say Partisan->Quantity
   @19,26 Say Partisan->Unit

   * Determine If memo field has an entry
   lMemo := MemoStatus(Partisan->Remarks)
   @ 11, 68 CheckBox lMemo

Return .T.


* ============================================================
Function Pro_Loc( cLocation, cDistrict, nDisRow, nDisCol, cProvince, ;
          nProRow, nProCol )
* ============================================================
Local cAlias := Select()

   If Seek_Rel( cLocation, "Location", 1 )
      If !Empty( cProvince )
      cProvince := Location->Province
      EndIf
      cDistrict := Location->District
      Select Location
      SetOrder( "Location" )
      Lookup( cLocation, "Location", "Proper(District)", nDisRow, nDisCol )
      If !Empty( nProRow )
      Lookup( cLocation, "Location", "Proper(Province)", nProRow, nProCol )
      EndIf
   Else
```

```
      Pop_Msg( "cLocation not found" )
   EndIf
   Select( cAlias )
Return .T.

*  ═══════════════════════════════════════════════════════════════
Function Ple_Scr
*  ═══════════════════════════════════════════════════════════════

   SysColor( AACLR_BOX )
   Boxer( 3, 7, 23, 73, 2,,, "GR+/B" ,, .T. )
   @ 8, 7 Say "╟"
   @ 8,73 Say "╢"
   @12, 7 Say "╟"
   @12,73 Say "╢"
   @17, 7 Say "╟"
   @17,73 Say "╢"
   @19,73 Say "╢"
   @23,39 Say "╧"
   @23,52 Say "╨"
   @ 8, 8 Say "──────────────────────────────────────────────────────────────"
   @12, 8 Say "──────────────────────────────────────────────────────────────"
   @17, 8 Say "──────────────────────────────────────────────────────────────"
   @18,39 Say "│   Date      ╟      Entry Type"
   @19, 9 Say Replicate(" ", 17 - ;
              Len( Trim( SubStr( Defaults->CarryOv_Hd, 1, 17 )))) ;
              Proper( Trim( SubStr( Defaults->CarryOv_Hd, 1, 17 ) )      :"
   @19,39 Say "├──────────────╫──────────────────"
   @20, 9 Say Replicate(" ", 17 - ;
              Len( Trim( SubStr( Defaults->Cur_Com_Hd, 1, 17 )))) + ;
              Proper( Trim( SubStr( Defaults->Cur_Com_Hd, 1, 17 ))) + ":"
   @20,39 Say "│              ╟        Donor:"
   @21, 9 Say Replicate(" ", 17 - ;
              Len( Trim( SubStr( Defaults->Nex_Com_Hd, 1, 17 )))) + ;
              Proper( Trim( SubStr( Defaults->Nex_Com_Hd, 1, 17 ))) + ":"
   @21,39 Say "│  CHAP FY    ╟      Channel:"
   @22, 9 Say Replicate(" ", 17 - ;
              Len( Trim( SubStr( Defaults->Disburs_Hd, 1, 17 )))) + ;
              Proper( Trim( SubStr( Defaults->Disburs_Hd, 1, 17 ))) + ":"
   @22,39 Say "│              ╟  Implementor:"

   SysColor( AACLR_BOX )
   @ 4,62 Say "Memo:"
   @ 5,12 Say "Pledge No:                  Donor/NGO Project No:"
   @ 6,10 Say "Activity No:                          Entry Date:"
   @ 7,13 Say "Activity:                                   dd/mm/yy"
   @ 9,10 Say "Organization:                       Donor:"
   @10,12 Say "Implementing Agency:"
   @11,10 Say "Organization Contacts:"
   @13,23 Say "Location:"
   @14,10 Say "Type of Beneficiaries:"
   @15,11 Say "No. of Beneficiaries:"
   @16,11 Say "Activity Life - From:          To           (dd/mm/yy)"

   SysColor( AACLR_HEADER )
   @ 3,20 Clear To  3,60
   @ 3,21 Say "P L E D G E S  /  C O M M I T M E N T S"
   SysColor( 1 )

Return .T.

*  ═══════════════════════════════════════════════════════════════
Function Ple_Key
*  ═══════════════════════════════════════════════════════════════
```

```
MemVar mPermission, mPledge, mEntry_Date, mPledge_No, mActNo

Local lResult     := .F.
Local Flds_1      := { "ActNo", "Proper(TITLE)" },;        // Activity array
      Act_Header  := { "Activity:Number", "Activity:Title" } // Activity header
Local cActFilter
Local GetList     := {}

   SysColor(2)     // Gets

   If mPermission <> "EXPERT"
      Return .F.
   EndIf

   If M->Is_Append
      mPledge_No    := "new "
      mEntry_Date := Date()
   EndIF

   @ 5,23 Say mPledge_No

   cActFilter := "Len( Trim( ActTitle->ActNo ) ) == 6 "
   @ 6,23 Get mActNo BOX B_SINGLE;
              Picture "9.9.99" ;
              Valid Help_Fill( @mActNo, "ActTitle", Flds_1 ;
                       ,, cActFilter,,,,, Act_Header,,, 1, 1 ) .and. ;
                Lookup( mActNo, "ActTitle", "Proper(Title)", 7, 23 )

   @ 6,60 Get mEntry_Date Picture "@D"

   lResult := Read_It( ,,GetList )

   SysColor(1)          // Normal

Return lResult


* ============================================================
Function Ple_Fld
* ============================================================
MemVar mEntry_Type, mPledge_No, mDonor, mOrganizat, mHumanAs_FY

Local lResult       := .F.
Local lDonor_Toggle := .F., ;
      lChann_Toggle := .F., ;
      lImple_Toggle := .F.

Local GetList := {}

   SysColor(2)     // Gets

   If M->Is_Append
      mEntry_Type := "   "
      mHumanAs_FY := Defaults->HumanAs_FY
   Else
      mHumanAs_FY := Pledge->HumanAs_FY
      mEntry_Type := Pledge->Entry_Type
      lDonor_Toggle := lIF( 'D' $ Pledge->Entry_Type, .T., .F. )
      lChann_Toggle := lIF( 'C' $ Pledge->Entry_Type, .T., .F. )
      lImple_Toggle := lIF( 'I' $ Pledge->Entry_Type, .T., .F. )
   EndIF

   @ 5,60 Get mProject_No Picture "@!S12"

   @ 9,24 Get mOrganizat BOX B_SINGLE;
                         Picture CAPS ;
```

```
                    Valid GetOrganizat( @mOrganizat )

   @ 9,53 Get mDonor BOX B_SINGLE;
                  Picture CAPS ;
                  Valid GetDonor( @mDonor )

   @10,33 Get mAgency Picture CAPS

   @11,33 Get mContact Picture CAPS
   @13,33 Get mLocation Picture CAPS
   @14,33 Get mClassBenef Picture CAPS
   @15,33 Get mNo_Benefic Picture "999,999,999"
   @16,33 Get mFrom Picture "@D"
   @16,47 Get mTo Picture "@D"
   @19,28 Get mCarryOver Picture "999,999,999"
   @20,28 Get mCommitment Picture "999,999,999"
   @21,28 Get mOtherComit Picture "999,999,999"
   @22,28 Get mDis_Amnt Picture "999,999,999"

   @20,42 Get mPledgeDate Picture "@D"
   @22,43 GET mHUMANAS_FY Picture "99/99"

   @20,68 CheckBox lDonor_Toggle
   @21,68 CheckBox lChann_Toggle
   @22,68 CheckBox lImple_Toggle Valid GetEntry( @mEntry_Type, ;
                           lDonor_Toggle, lChann_Toggle, ;
                           lImple_Toggle )

   lResult := Read_It( ,,GetList )

   If lResult
      If M->Is_Append
      @ 5,23 Say ( mPledge_No := Next( "Pledge_No" )) Picture "@9"
      Else
      If UpDated()
        Aud_After()          // Check Audit Trail for changes
      EndIf
      EndIf
   EndIf

Return lResult


* ============================================================
Function Ple_Say
* ============================================================
Local lMemo     := .F.
Local GetList := {}
Local lDonor_Toggle := IIF( 'D' $ Pledge->Entry_Type, .T., .F. )
Local lChann_Toggle := IIF( 'C' $ Pledge->Entry_Type, .T., .F. )
Local lImple_Toggle := IIF( 'I' $ Pledge->Entry_Type, .T., .F. )

   SysColor( 2 )                                    // hilighted video
   * Set Keys for tagging and untagging
   Set Key K_ALT_A To Tag_Record           // Menu to Tag or Untag all
   Set Key K_ALT_T To Tag_Record           // Tag a record
   Set Key K_ALT_U To Tag_Record                  // Untag a record
   Set Key K_ALT_X To Imm_Exit             // Exit from program

   SYSCOLOR ( "N/W" )

   @ 5,23 Say Transform( Val( Pledge->Pledge_No), "@Z" )
   @ 5,60 Say Pledge->Project_No
   @ 6,23 Say Pledge->ActNo
   @ 7,23 Say SubStr(Proper( Fetch( Pledge->ActNo, "ActTitle", "Title", ;
                              "ActNo" ) ), 1, 36 )
```

```
   @ 6,60 Say Pledge->Entry_Date
   @ 9,24 Say Pledge->Organizat
   @ 9,53 Say SubStr( Pledge->Donor, 1, 19 )
   @10,33 Say Pledge->Agency
   @11,33 Say Proper( Pledge->Contact )
   @13,33 Say Proper( Pledge->Location )
   @14,33 Say Proper( Pledge->ClassBenef )
   @15,33 Say Pledge->No_Benefic Picture "999,999,999"
   @16,33 Say Pledge->From Picture "@D"
   @16,47 Say Pledge->To Picture "@D"
   @19,28 Say Pledge->CarryOver Picture "999,999,999"
   @20,28 Say Pledge->Commitment Picture "999,999,999"
   @21,28 Say Pledge->OtherComit Picture "999,999,999"
   @22,28 Say Pledge->Dis_Amt Picture   "999,999,999"
   @20,42 Say Pledge->PledgeDate Picture "@D"
   @22,43 SAY PLEDGE->HUMANAS_FY

   @20,68 CheckBox lDonor_Toggle
   @21,68 CheckBox lChann_Toggle
   @22,68 CheckBox lImple_Toggle

   lMemo := MemoStatus(Pledge->Remarks)
   @ 01, 68 CheckBox lMemo

Return .T.


* ============================================================
Function GetEntry( mEntry_Type, lDonor, lChann, lImple )
* ============================================================
Local lResult := .F.
   mEntry_Type := IIF( lDonor, "D ", "  " )
   mEntry_Type := SubStr(mEntry_Type,1,1) + IIF( lChann, "C ", "  " )
   mEntry_Type := SubStr(mEntry_Type,1,2) + IIF( lImple, "I", " " )

   If mEntry_Type == "   "
      Pop_Msg( "Donor, Channel and Implementor may not all be blank." )
   Else
      lResult := .T.
   EndIf

Return lResult


* ============================================================
Function GetDonor( cDonor )
* ============================================================
Local lResult := .F.
Local Flds_10    := { "Donor", "Descript" }, ; // Donor array
      Don_Header := { "Donor", "Description" }, ; // Donor header
      Flds_3     := { "Agency", "Proper(AgencyLong)" }, ;     // Agency array
      Age_Header := { "Donor Agency", "Agency Description" }

Local cSaveFile   := Select()
Local cSaveRecNo  := RecNo()

   If Seek_Rel( cDonor, "Donor" )
      lResult := .T.
   Else
      If Seek_Rel( cDonor, "Agency" )
        lResult := .T.
      EndIf
   EndIf

   If !lResult
      lResult := Help_Fill( @cDonor, "Donor", Flds_10 ;
```

```
                              ,,,,,,, Don_Header,,, 1, 1 )
      If Trim( cDonor ) == "AGENCY" .or. cDonor == Space( 20 )
        lResult := Help_Fill( @cDonor, "Agency", Flds_3 ;
                              ,,,,,,, Age_Header,,, 1, 1 )
      EndIf
   EndIf

   Select( cSaveFile )
   Go cSaveRecNo
Return lResult


* ============================================================
Function GetOrganizat( cOrganizat )
* ============================================================
Local lResult := .F.
Local Flds_10     := { "Donor", "Descript" }, ; // Donor array
      Don_Header  := { "Organization", "Description" }, ; // Donor header
      Flds_3      := { "Agency", "Proper(AgencyLong)" }, ;     // Agency array
      Age_Header  := { "Organization:Agency", "Organization:Description" }

Local cSaveFile   := Select()
Local cSaveRecNo  := RecNo()

   If Seek_Rel( cOrganizat, "Donor" )
      lResult := .T.
   Else
      If Seek_Rel( cOrganizat, "Agency" )
        lResult := .T.
      EndIf
   EndIf

   If !lResult

      lResult := Help_Fill( @cOrganizat, "Donor", Flds_10 ;
                            ,,,,,,, Don_Header,,, 1, 1 )
      If Trim( cOrganizat ) == "AGENCY" .or. cOrganizat == Space( 20 )
        lResult := Help_Fill( @cOrganizat, "Agency", Flds_3 ;
                              ,,,,,,, Age_Header,,, 1, 1 )
      EndIf
   EndIf

   Select( cSaveFile )
   Go cSaveRecNo

Return lResult


* ============================================================
Function Act_Scr
* ============================================================

   SysColor( AACLR_BOX )
   Boxer( 4, 2, 23, 77, 2,,, "GR+/B" ,, .T. )
   @ 9, 2 Say "├"
   @ 9,77 Say "┤"
   @15, 2 Say "├"
   @15,77 Say "┤"
   @19, 2 Say "├"
   @19,77 Say "┤"
@ 9, 3 Say "──────────────────────────────────────────────────────────────────────────"
@15, 3 Say "──────────────────────────── Objective ─────────────────────────────────"
@19, 3 Say "──────────────────────── Summary (English) ─────────────────────────────"

   SysColor( AACLR_NORMAL )
   @ 6, 5 Say "Activity No:"
```

```
   @ 6,54 Say Replicate( " ", 10 - ;
              Len(Trim(SubStr(Defaults->Other_Lang, 1, 10)))) + ;
              Proper(Trim(SubStr(Defaults->Other_Lang, 1, 10))) + " Summary:"
   @ 7, 5 Say "Activity Title:"
   @ 8, 5 Say "Long Title "
   @10, 5 Say "Target Population:"
   @11, 5 Say "Concerned Agencies:"
   @12, 5 Say "Donors "
   @13, 5 Say "Time Frame:"
   @14, 5 Say "Dollar Needs:                                  Total Commitments:"

   SysColor( AACLR_HEADER )
   @ 4,17 Clear To  4,63
   @ 4,18 Say "C O N S O L I D A T E D   A C T I V I T I E S"

   SysColor( AACLR_NORMAL )

Return .T.


* ============================================================
Function Act_Key
* ============================================================
MemVar mPermission, mTag, mActNo

Local lResult     := .F.
Local Flds_1      := { "ActNo", "Proper(Title)" },;       // Activity array
      Act_Header  := { "Activity:Number", "Activity:Title" }  //Activity header
Local cActFilter
Local GetList     := {}

   SysColor(2)     // Gets

   If mPermission <> "EXPERT"
      Return .F.
   EndIf

   mTag := .T.             // TAG true If edit or add

   cActFilter := "Len( Trim( ActTitle->ActNo ) ) == 6 "
   @ 6,21 Get mActNo BOX B_SINGLE;
              Picture "9.9.9X" ;
              Valid Help_Fill( @mActNo, "ActTitle", Flds_1 ;
                    ,, cActFilter,,,,, Act_Header,,, 1, 1 ) .and. ;
                    Unique( mActNo ) .and. ;
                Lookup( mActNo, "ActTitle", "Proper(Title)", 7, 21 );
                .and. ;
                Lookup( mActNo, "ActTitle", ;
                     "Proper(SubStr(Long_Title, 1, 55))", ;
                     8, 21 )

   SysColor(1)          // Normal

Return Read_It( ,,GetList)


* ============================================================
Function Act_Fld
* ============================================================
MemVar mPermission, mDonors

Local lResult     := .F.
Local GetList     := {}

   SysColor(2)     // Gets
```

```
   If M->Is_Append
      Aud_Before()
   Endif

   If mPermission <> "EXPERT"
      Return .F.
   EndIf

   @10,25 Get mTarget_Pop Picture "@S51"
   @11,25 Get mAgencies Picture   "@S51"

   // Updates donor field from donors in pledge file and paints to screen
   // parameters: cDonors, nRow, nCol, lUpdate, lPaint, lDonorOnly ;
   // nDisplayLength, .
   //             nDonorFieldLen
   @12,25 Get mDonors Picture "@S51" ;
                  Valid GetActDonors( @mDonors, 12, 25, .T., .T., .T., ;
                                      51, 126 )

   @13,25 Get mTime_Frame Picture "@S51"

   lResult := Read_It( ,,GetList )

   If lResult
      If M->Is_Append
      Else
      If UpDated()
        Aud_After()          // Check Audit Trail for changes
      EndIf
      EndIf
   EndIf

Return lResult


* ============================================================
Function Act_Say
* ============================================================
Local nDolNeeds := 0, nTot_Commit := 0, nActNo := Activity->ActNo
Local nCarry    := 0, nComm  := 0
Local cOld_Help := HelpCode()
Local lMemo     := .F.
Local GetList := {}

   SysColor( 2 )                                  // hilighted video
   * Set Keys for tagging and untagging
   Set Key K_ALT_A To Tag_Record          // Menu to Tag or Untag all
   Set Key K_ALT_T To Tag_Record          // Tag a record
   Set Key K_ALT_U To Tag_Record                 // Untag a record
   Set Key K_ALT_X To Imm_Exit            // Exit from program

   * Determine If memo field has an entry
   lMemo := MemoStatus( "Summary_Po" )
   @ 06, 73 CheckBox lMemo

   SYSCOLOR ( "N/W" )

   Select Needs
   If Seek_Rel( nActNo, "Needs" )
      Sum Needs->Amount To nDolNeeds While Needs->ActNo = nActNo
   Else
      IIF( SubStr(Activity->ActNo,5,2) = '00', '', ;
       Pop_Msg( "Unable to find " + Activity->ActNo + " in NEEDS" ) )
   EndIf

   Select Pledge
```

```
   If Seek_Rel( nActNo, "Pledge" )
      Sum Pledge->CarryOver, Pledge->Commitment To nCarry, nComm ;
       While Pledge->ActNo = nActNo For 'D' $ Pledge->Entry_Type
      nTot_Commit := nCarry + nComm
   Else
      nTot_Commit := 0
   EndIf

   Select Activity

   @ 6,21 Say Activity->ActNo
   @ 7,21 Say Proper( ActTitle->Title )
   @ 8,21 Say SubStr( Proper( ActTitle->Long_Title ), 1, 55)
   @10,25 Say SubStr( Proper( Activity->Target_Pop ), 1, 51)
   @11,25 Say SubStr( Activity->Agencies, 1, 51)

   // parameters: cDonors, nRow, nCol, lUpdate, lPaint, lDonors Only:
   // nDisplayLength, nDonorFieldLen
   GetActDonors( Activity->Donors, 12, 25, .F., .T., .T., 51, 126 )

   @13,25 Say SubStr( Proper( Activity->Time_Frame ), 1, 51)
   @14,25 Say nDolNeeds Picture "999,999,999"
   @14,65 Say nTot_Commit Picture "999,999,999"


   KeyBoard CHR( K_ESC )
   MemoEdit(Activity->Objective, 16, 05, 17, 74, .F.)

   KeyBoard CHR( K_ESC )
   MemoEdit(Activity->Summary, 20, 05, 22, 74, .F.)

   HelpCode( cOld_Help )

Return .T.

*  ---------------------------------------------------------------------------
Function Inc_Scr
*  ---------------------------------------------------------------------------
   Local SubHeader := "GR+/B"

   SysColor( AACLR_BOX )
   Boxer( 2, 0, 23, 79, 2,,, "GR+/B" ,, .T. )
   @ 6, 0 Say "╟"
   @ 6,79 Say "╢"
   @17, 0 Say "╟"
   @17,79 Say "╢"
   @ 6, 1 Say
"──────────────────────────────────────────────────────────────────────────────"
   @17, 1 Say
"──────────────────────────────────────────────────────────────────────────────"

   SysColor( AACLR_BOX )
   @ 3,62 Say "dd/mm/yy"
   @ 4, 3 Say "Incident No:                                  Date of Incident:"
   @ 5, 8 Say "Source:                                     Type of Incident:"
   @ 8, 4 Say "Location:"
   @ 9, 2 Say  Replicate( " ", 10 - ;
              Len(Trim(SubStr(Defaults->District, 1,10)))) + ;
              Proper(Trim(SubStr(Defaults->District, 1, 10 ))) + ":"
   @10, 2 Say  Replicate( " ", 10 - ;
              Len(Trim(SubStr(Defaults->Province, 1,10)))) + ;
              Proper(Trim(SubStr(Defaults->Province, 1, 10))) + ":"
   @12, 3 Say "Map Sheet:"
   @13, 4 Say "Grid Ref:"
   @14, 4 Say "Latitude:"
```

```
   @15, 3 Say "Longitude:"
   @19, 8 Say "Dead:"
   @20, 5 Say "Injured:"
   @21, 3 Say "Kidnapped:"

   SysColor( AACLR_HEADER )

   @ 2,25 Clear To  2,55
   @ 2,26 Say "I N C I D E N T   R E P O R T"
   Boxer( 7, 37, 11, 77, 4,,, "W+/B" ,, .F. )
   @ 7,47 Clear To  7,68
   Boxer( 12, 37, 16, 77, 4,,, "W+/B" ,, .F. )
   @12,47 Clear To 12,68
   Boxer( 18, 20, 22, 47, 4,,, "W+/B" ,, .F. )
   @18,23 Clear To 18,41
   Boxer( 18, 50, 22, 77, 4,,, "W+/B" ,, .F. )
   @18,54 Clear To 18,73
   @ 7,48 Say "Incident Description"
   @12,48 Say "Location Description"
   @18,24 Say "Casualty Description"
   @18,55 Say "Damage Description"

Return .T.


* ============================================================
Function Inc_Key
* ============================================================
MemVar mPermission, mTag, mInc_Number, mEntry_Date, mInc_Date

Local lResult   := .F.
Local GetList := {}

   SysColor(2)      // Gets

   If mPermission <> "EXPERT"
      Return .F.
   EndIf

   mTag := .T.              // TAG true If edit or add

   If M->Is_Append
      mInc_Number := "new "
      mInc_Date   := Date()
      mEntry_Date := Date()
   Else
   EndIF

   @ 4,16 Say mInc_Number
   @ 4,62 Get mInc_Date ;
              Picture "@D" ;
              Valid mInc_Date <= Date() .and. ;
                    mInc_Date >= Ctod( "01/10/92" )

   IF lResult := Read_It( ,,GetList )
      If M->Is_Append
      Else
      EndIf
   EndIf

   SysColor(1)          // Normal

Return lResult

* ============================================================
```

```
Function Inc_Fld
*
MemVar mDistrict, mLatitude, mLongitude, mPermission, mInc_Type
MemVar mLocation, mInc_Number

Local lResult := .F.

Local Flds_2      := { "Location", "District", "Province" }, ; // Location array
      Loc_Header  := { "Location",  Proper(Trim(Defaults->District)), ,
                       Proper(Trim(Defaults->Province)) }     // Location header
Local Flds_6      := { "Alias", "Inc_Type" }, ; // Incident Type array
      Inc_Header  := { "Incident Type,Short Version", "Incident Type" }
Local Flds_5      := { "District", "Province" }, ; // District array
      Dis_Header  := {  Proper(Trim(Defaults->District)), ,
                        Proper(Trim(Defaults->Province)) } // District header

Local cProvince := Space( 20 )
Local lMemo     := .F.
Local LocFilter := "Location->Location != ' ' "
Local cNS_Lat   := "S", ;
      cEW_Long  := "E"

Local GetList := {}

   SysColor(2)     // Gets

   If M->Is_Append
      mDistrict := Space( 20 )
      mLatitude  := 0.00
      mLongitude := 0.00
      cNS_Lat    := "S"
      cEW_Long   := "E"
   Else
      mLatitude  := LatDeg( Incident->Latitude, @cNS_Lat )
      mLongitude := LongDeg( Incident->Longitude, @cEW_Long )
   Endif

   If mPermission <> "EXPERT"
      Return .F.
   EndIf


   @ 5,16 Get mRep_Name Picture CAPS

   @ 5,62 Get mInc_Type BOX B_SINGLE;
              Picture CAPS ;
              Valid Help_Fill( @mInc_Type, "Inc_Type", Flds_6 ;
                       ,,,,,,, Inc_Header,,, 1, 1 )

   @ 8,14 Get mLocation ;       // BOX B_SINGLE;
              Picture CAPS ;
              Valid ( Empty( mLocation ) .or. ;
                   ( Help_Fill( @mLocation, "Location", Flds_2 ;
                       ,, LocFilter,,,,, Loc_Header, ;
                       ( !! GetOrder("LOCATION") ) ;
                       ,, 1, 1 ) .and. ;
             Pro_Loc( mLocation, @mDistrict, 9, 14, cProvince, 10, 14 ) ) ) ;
               HELP "Any key + <ENTER> for picklist"

   @ 9,14 Get mDistrict BOX B_SINGLE;
              Picture CAPS ;
              When Empty( mLocation ) ;
              Valid Help_Fill( @mDistrict, "Location", Flds_5 ;
                       ,,,,,,, Dis_Header, ;
                       ( !! GetOrder("District") ) ;
                       ,, 1, 2 ) .and. ;
```

```
          Pro_Dist( mDistrict, 10, 14 ) // paint Province at 10,14

   @12,14 Get mMap_Sheet Picture CAPS
   @13,14 Get mGrid_Ref Picture CAPS

   @14,15 Get mLatitude Picture "@R  99° 99.99'" ;
            Valid mLatitude >= 0 .and. mLatitude <= 9000
   @14,Col() + 1 Get cNS_Lat Picture CAPS ;
               Valid cNS_Lat $ "NS"

   @15,14 Get mLongitude Picture "@R  999° 99.99'" ;
            Valid mLongitude >= 0 .and. mLongitude <= 18000
   @15,Col() + 1 Get cEW_Long Picture CAPS ;
               Valid cEW_Long $ "EW"


   @19,14 Get mDead Picture [9999]
   @20,14 Get mInjured Picture [9999]
   @21,14 Get mKidnapped Picture [9999]

   lResult := Read_It( ,,GetList )

   If lResult
      mLatitude  := LatDec( mLatitude, cNS_Lat )
      mLongitude := LongDec( mLongitude, cEW_Long )

      If M->Is_Append
        @ 4,16 Say ( mInc_Number := Next( "Inc_Number" )) Picture "@9"
      EndIf
   EndIf

Return lResult


* ============================================================
Function Inc_Say
* ============================================================

Local lMemo     := .F.
Local GetList := {}

   SysColor( 2 )                                  // hilighted video
   * Set Keys for tagging and untagging
   Set Key K_ALT_A To Tag_Record          // Menu to Tag or Untag all
   Set Key K_ALT_T To Tag_Record          // Tag a record
   Set Key K_ALT_U To Tag_Record                  // Untag a record
   Set Key K_ALT_X To Imm_Exit            // Exit from program

   SysColor ( "N/W" )

   @ 4,16 Say Transform( Val( Incident->Inc_Number), "@Z" )
   @ 5,16 Say Incident->Rep_Name

   @ 4,62 Say Incident->Inc_Date
   @ 5,62 Say Incident->Inc_Type

   @ 8,14 Say Proper( Incident->Location )
   @ 9,14 Say Proper( Incident->District )
   @10,14 Say Proper( Fetch( Incident->District, "Location", "Province", 2 ) )
   @12,14 Say Incident->Map_Sheet
   @13,14 Say Incident->Grid_Ref

   DispLatLong( Incident->Latitude, 14, 15, Incident->Longitude, 15, 14 )

   @19,14 Say Incident->Dead
   @20,14 Say Incident->Injured
```

```
   @21,14 Say Incident->Kidnapped

   SysColor( "N/W" )
   KeyBoard CHR( K_ESC )
   MemoEdit(Incident->Inc_Desc, 08, 39, 10, 75, .F.)

   KeyBoard CHR( K_ESC )
   MemoEdit(Incident->Loc_Desc, 13, 39, 15, 75, .F )

   KeyBoard CHR( K_ESC )
   MemoEdit(Incident->Inj_Desc, 19, 22, 21, 45, .F.)

   KeyBoard CHR( K_ESC )
   MemoEdit(Incident->Dam_Desc, 19, 52, 21, 75, .F )

Return .T.

*  ============================================================================
Function Mai_Scr
*  ============================================================================
   Boxer( 2, 2, 23, 77, 2,,, "GR+/B" ,, .F. )

   SysColor( AACLR_HEADER )
   @ 2,24 Clear To  2,48
   @ 2,25 Say "M A I L I N G   L I S T"

   Boxer( 3, 5, 9, 45, 1,,, "W+/B" ,, .T. )
   SysColor( AACLR_BOX )
   @ 4, 8 Say "Honorific:"
   @ 5,12 Say "Title:"
   @ 6, 7 Say "First Name:"
   @ 7, 8 Say "Last Name:"
   @ 8,09 Say "Position:"

   SysColor( AACLR_HEADER )
   @ 3, 7 Clear To  3,12
   @ 3, 8 Say "NAME"

   Boxer( 3, 50, 8, 74, 1,,, "W+/B" ,, .T. )
   SysColor( AACLR_BOX )
   @ 5,52 Say "OTel:"
   @ 6,52 Say "HTel:"
   @ 7,53 Say "Fax:"

   SysColor( AACLR_HEADER )
   @ 3,52 Clear To  3,67
   @ 3,53 Say "COMMUNICATIONS"

   Boxer( 11, 5, 21, 57, 1,,, "W+/B" ,, .T. )
   SysColor( AACLR_BOX )
   @13, 7 Say "Organization:"
   @14,10 Say "Address 1:"
   @15,10 Say "Address 2:"
   @16,15 Say "City:"
   @17, 9 Say Replicate( " ", 10 - Len(Trim(Defaults->Province))) + ;
          Proper(Trim(Defaults->Province))+":"
   @18,12 Say "Country:"
   @20,07 Say "Type of Organization:"

   SysColor( AACLR_HEADER )
   @11, 7 Clear To 11,20
   @11, 8 Say "ORGANIZATION"

   Boxer( 11, 61, 21, 74, 1,,, "W+/B" ,, .T. )
   SysColor( AACLR_BOX )
   @13,62 Say Replicate( " ",6 - ;
```

```
            Len( Trim( SubStr( Defaults->Mail_List1, 1,6 )))) + ;
            Proper( Trim( SubStr( Defaults->Mail_List1, 1,6))) + ":"
   @14,62 Say Replicate( " ",6 - ;
            Len( Trim( SubStr( Defaults->Mail_List2, 1,6 )))) + ;
            Proper( Trim( SubStr( Defaults->Mail_List2, 1,6))) + ":"
   @15,62 Say Replicate( " ",6 - ;
            Len( Trim( SubStr( Defaults->Mail_List3, 1,6 )))) + ;
            Proper( Trim( SubStr( Defaults->Mail_List3, 1,6))) + ":"
   @16,62 SAY Replicate( " ",6 - ;
            Len( Trim( SubStr( Defaults->Mail_List4, 1,6 )))) + ;
            Proper( Trim( SubStr( Defaults->Mail_List4, 1,6))) + ":"
   @17,62 SAY Replicate( " ",6 - ;
            Len( Trim( SubStr( Defaults->Mail_List5, 1,6 )))) + ;
            Proper( Trim( SubStr( Defaults->Mail_List5, 1,6))) + ":"
   @18,62 SAY Replicate( " ",6 - ;
            Len( Trim( SubStr( Defaults->Mail_List6, 1,6 )))) + ;
            Proper( Trim( SubStr( Defaults->Mail_List6, 1,6))) + ":"
   @19,62 SAY Replicate( " ",6 - ;
            Len( Trim( SubStr( Defaults->Mail_List7, 1,6 )))) + ;
            Proper( Trim( SubStr( Defaults->Mail_List7, 1,6))) + ":"
   @20,62 SAY Replicate( " ",6 - ;
            Len( Trim( SubStr( Defaults->Mail_List8, 1,6 )))) + ;
            Proper( Trim( SubStr( Defaults->Mail_List8, 1,6))) + ":"

   SysColor( AACLR_HEADER )
   @11,63 Clear To 11,69
   @11,64 Say "LISTS"

Return .T.


* ============================================================
Function Mai_Key
* ============================================================
MemVar mTag, mTitle, mHonorific, mPermission, mFirst_Name, mLast_Name

Local lResult := .F.
Local Flds_1     := { "Honorific" }, ;  // Honorific array
      Hon_Header := { "Honorific" }
Local Flds_2     := { "Title" }, ;      // Title array
      Tit_Header := { "Title" }

Local GetList := {}

   SysColor(2)     // Gets
   If mPermission <> "EXPERT"
      Return .F.
   EndIf

   mTag := .T.              // TAG true If edit or add

   If M->Is_Append
      // default values for an add or dup
      mTitle     := "Mr.           "
      mHonorific := "              "
   EndIF

   @ 4,19 Get mHonorific Valid Empty( mHonorific ) .or. ;
                         Help_Fill( @mHonorific, "Honorif", Flds_1 ;
                                    ,,,,,,, Hon_Header,,,1 ,1 )
                         HELP "Any key + <ENTER> for picklist"

   @ 5,19 Get mTitle Valid Empty( mTitle ) .or. ;
                     Help_Fill( @mTitle, "Title", Flds_2 ;
                                ,,,,,,, Tit_Header,,,1 ,1 )
                     HELP "Any key + <ENTER> for picklist"
```

```
   @ 6,19 Get mFirst_Name
   @ 7,19 Get mLast_Name

   lResult := Read_It( ,,GetList )

   SysColor(1)          // Normal

Return lResult



* ============================================================
Function Mai_Fld
* ============================================================
MemVar mPermission, mClassify, mMail_List, mTag, mPosition, mOrgar .it
MemVar mAddress1, mAddress2, mCity, mProvince, mCountry, mPhone_1
MemVar mPhone_2, mFax_1

Local lResult := .F.
Local lMemo   := .F.
Local GetList := {}

Local Flds_1     := { "Alias", "Proper(Classify)" }, ; // Classification array
      Cla_Header := { "Code", "Classification" }

Local lM1_Toggle := .F., ;
      lM2_Toggle := .F., ;
      lM3_Toggle := .F., ;
      lM4_Toggle := .F., ;
      lM5_Toggle := .F., ;
      lM6_Toggle := .F., ;
      lM7_Toggle := .F., ;
      lM8_Toggle := .F.

   SysColor(2)     // Gets

   If M->ls_Append
   Else
      lM1_Toggle := IIF( '1' $ Mail->Mail_List, .T., .F. )
      lM2_Toggle := IIF( '2' $ Mail->Mail_List, .T., .F. )
      lM3_Toggle := IIF( '3' $ Mail->Mail_List, .T., .F. )
      lM4_Toggle := IIF( '4' $ Mail->Mail_List, .T., .F. )
      lM5_Toggle := IIF( '5' $ Mail->Mail_List, .T., .F. )
      lM6_Toggle := IIF( '6' $ Mail->Mail_List, .T., .F. )
      lM7_Toggle := IIF( '7' $ Mail->Mail_List, .T., .F. )
      lM8_Toggle := IIF( '8' $ Mail->Mail_List, .T., .F. )
      Aud_Before()
   Endif

   If mPermission <> "EXPERT"
      Return .F.
   EndIf

   @ 8,19 Get mPosition  Picture "@S25"
   @13,21 Get mOrganizat  Picture "@S35"
   @14,21 Get mAddress1
   @15,21 Get mAddress2
   @16,21 Get mCity  Picture CAPS
   @17,21 Get mProvince  Picture CAPS
   @18,21 Get mCountry  Picture CAPS

   @20,29 Get mClassify BOX B_SINGLE;
              Picture CAPS ;
              Valid Help_Fill( @mClassify, "Classify", Flds_1 ;
                       ,,,,,,, Cla_Header,,,1 ,1 ) .And. .
```

```
            LookUp( mClassify, "Classify", "Classify", 20, 29 )

   @ 5,58 Get mPhone_1
   @ 6,58 Get mPhone_2
   @ 7,58 Get mFax_1

   @13,70 CheckBox lM1_Toggle
   @14,70 CheckBox lM2_Toggle
   @15,70 CheckBox lM3_Toggle
   @16,70 CheckBox lM4_Toggle
   @17,70 CheckBox lM5_Toggle
   @18,70 CheckBox lM6_Toggle
   @19,70 CheckBox lM7_Toggle
   @20,70 CheckBox lM8_Toggle Valid GetMList( @mMail_List, ;
                                 lM1_Toggle, lM2_Toggle, lM3_Togg.e, ;
                                 lM4_Toggle, lM5_Toggle, lM6_Toggle, ;
                                 lM7_Toggle, lM8_Toggle ;
                                 )

   lResult := Read_It( ,,GetList )

   If lResult
      mTag := .T.
      If M->Is_Append
      Else
      If UpDated()
        Aud_After()          // Check Audit Trail for changes
      EndIf
      EndIf
   EndIf

Return lResult


* ============================================================================
Function Mai_Say
* ============================================================================
Local lMemo     := .F.
Local GetList := {}

Local lM1_Toggle := IIF( '1' $ Mail->Mail_List, .T., .F. )
Local lM2_Toggle := IIF( '2' $ Mail->Mail_List, .T., .F. )
Local lM3_Toggle := IIF( '3' $ Mail->Mail_List, .T., .F. )
Local lM4_Toggle := IIF( '4' $ Mail->Mail_List, .T., .F. )
Local lM5_Toggle := IIF( '5' $ Mail->Mail_List, .T., .F. )
Local lM6_Toggle := IIF( '6' $ Mail->Mail_List, .T., .F. )
Local lM7_Toggle := IIF( '7' $ Mail->Mail_List, .T., .F. )
Local lM8_Toggle := IIF( '8' $ Mail->Mail_List, .T., .F. )

   SysColor ( "N/W" )
   @ 4,19 Say Proper( Mail->Honorific )
   @ 5,19 Say Proper( Mail->Title )
   @ 6,19 Say Proper( Mail->First_Name )
   @ 7,19 Say Mail->Last_Name
   @ 8,19 Say SubStr( Proper( Mail->Position ), 1, 25 )

   @13,21 Say SubStr( Mail->Organizat, 1, 35 )
   @14,21 Say Mail->Address1
   @15,21 Say Mail->Address2
   @16,21 Say Proper( Mail->City )
   @17,21 Say Proper( Mail->Province )
   @18,21 Say Proper( Mail->Country )
   @20,29 Say Fetch( Mail->Classify, "Classify", "Classify" )

   @ 5,58 Say Mail->Phone_1
   @ 6,58 Say Mail->Phone_2
```

```
   @ 7,58 Say Mail->FAX_1

   @13,70 CheckBox IM1_Toggle
   @14,70 CheckBox IM2_Toggle
   @15,70 CheckBox IM3_Toggle
   @16,70 CheckBox IM4_Toggle
   @17,70 CheckBox IM5_Toggle
   @18,70 CheckBox IM6_Toggle
   @19,70 CheckBox IM7_Toggle
   @20,70 CheckBox IM8_Toggle

Return .T.


* ============================================================
Function GetMList( mMail_List, IM1, IM2, IM3, IM4, IM5, IM6, IM7, IM8 )
* ============================================================

   mMail_List := IIF( IM1, "1       ", "        " )
   mMail_List := SubStr(mMail_List,1,1) + IIF( IM2, "2", " " )
   mMail_List := SubStr(mMail_List,1,2) + IIF( IM3, "3", " " )
   mMail_List := SubStr(mMail_List,1,3) + IIF( IM3, "4", " " )
   mMail_List := SubStr(mMail_List,1,4) + IIF( IM3, "5", " " )
   mMail_List := SubStr(mMail_List,1,5) + IIF( IM3, "6", " " )
   mMail_List := SubStr(mMail_List,1,6) + IIF( IM3, "7", " " )
   mMail_List := SubStr(mMail_List,1,7) + IIF( IM3, "8", " " )

Return .T.


* ============================================================
Function Loc_Scr
* ============================================================
   Boxer( 3, 10, 21, 69, 2,,, "GR+/B" ,, .T. )

   SysColor( AACLR_BOX )
   @10,10 Say "├"
   @10,69 Say "┤"
   @15,10 Say "├"
   @15,69 Say "┤"

   @ 5,15 Say "Location:                              Location Code:"
   @ 6,13 Say  Replicate( " ",10 - ;
              Len(Trim(SubStr(Defaults->District, 1,10 )))) + ;
              Proper(Trim(SubStr(Defaults->District, 1,10))) + ":"
   @ 7,13 Say  Replicate( " ",10 - ;
              Len(Trim(SubStr(Defaults->Province, 1,10 )))) + ;
              Proper(Trim(SubStr(Defaults->Province, 1,10))) + ":"
   @ 8,19 Say "Type:"
   @10,11 Say "──────────────────────── ──────────────────────────────────────"
   @12,15 Say "Latitude:                    [  dd° mm.mm' (N/S) ]"
   @13,14 Say "Longitude:                    [ ddd° mm.mm' (E/W) ]"
   @15,11 Say "──────────────────────────────────────────────────────────────"

   SysColor( AACLR_HEADER )
   @ 3,32 Clear To  3,48
   @ 3,33 Say "L O C A T I O N"
   @10,28 Clear To 10,51
   @10,29 Say "Geographic Coordinates"
   @15,31 Clear To 15,48
   @15,32 Say "Location Remarks"

Return .T.


* ============================================================
```

```
Function Loc_Key
* ============================================================
Local lResult := .T.

Return lResult


* ============================================================
Function LOC_FLD
* ============================================================
MemVar mPermission, mTag, mLocation, mDistrict, mProvince, mType
MemVar mLatitude, mLongitude, mLoc_Code, mRemarks

Local lResult     := .F.
Local lMemo       := .F.
Local Flds_8      := { "Alias", "Proper(Loc_Type)" }, ; // Location Type array
      Typ_Header  := { "Code", "Location Type" }
Local cNS_Lat     := "S", ;
      cEW_Long    := "E"
Local GetList     := {}

   SysColor(2)     // Gets

   If mPermission <> "EXPERT"
      Return .F.
   EndIf

   mTag := .T.             // TAG true If edit or add

   If M->Is_Append
      mLocation  := Space( 20 )
      mDistrict  := Space( 20 )
      mProvince  := Space( 20 )
      mType      := "   "
      mLatitude  := 0.0
      mLongitude := 0.0
      mRemarks   := ""
      cNS_Lat    := "S"
      cEW_Long   := "E"
   Else
      mLatitude  := LatDeg( Location->Latitude, @cNS_Lat )
      mLongitude := LongDeg( Location->Longitude, @cEW_Long )
   EndIf

   @ 5,25 Get mLocation  Picture CAPS ;
                         Valid !Empty( mLocation )

   @ 6,25 Get mDistrict  Picture CAPS ;
                         Valid !Empty( mDistrict ) .and. ;
                               Unique( mDistrict + mLocation, 4 )
//                                     GetOrder( "District + Location" ) )

   @ 7,25 Get mProvince  Picture CAPS ;
                         Valid !Empty( mProvince )


   @ 8,25 Get mTYPE BOX B_SINGLE;
            Picture CAPS ;
            Valid ( Help_Fill( @mType, "Loc_Type", Flds_8 ;
                     ,,,,,,, Typ_Header,,, 1, 1 ) )

   @ 5,63 Get mLoc_Code  Picture CAPS

   @12,26 Say Space(11)
   @12,26 Get mLatitude Picture  "@R  99° 99.99'" ;
            Valid mLatitude >= 0 .and. mLatitude <= 9000
```

```
   @12,Col() + 1 Get cNS_Lat Picture CAPS ;
                 Valid cNS_Lat $ "NS"

   @13,25 Say Space(11)
   @13,25 Get mLongitude Picture "@R  999° 99.99'" ;
              Valid mLongitude >= 0 .and. mLongitude <= 18000
   @13,Col() + 1 Get cEW_Long Picture CAPS ;
                 Valid cEW_Long $ "EW"

   lResult := Read_It( ,,GetList )

   If lResult
      mTag := T
      mLatitude  := LatDec( mLatitude, cNS_Lat )
      mLongitude := LongDec( mLongitude, cEW_Long )
   EndIf

Return lResult


* ============================================================
Function Loc_Say
* ============================================================

Local lMemo    := .F
Local GetList := {}

   SysColor( 2 )                                  // hilighted video
   * Set Keys for tagging and untagging
   Set Key K_ALT_A To Tag_Record          // Menu to Tag or Untag all
   Set Key K_ALT_T To Tag_Record          // Tag a record
   Set Key K_ALT_U To Tag_Record                 // Untag a record
   Set Key K_ALT_X To Imm_Exit            // Exit from program

   SysColor ( "N/W" )

   @ 5,25 Say Proper( Location->Location )
   @ 6,25 Say Proper( Location->District )
   @ 7,25 Say Proper( Location->Province )

   @ 8,25 Say Proper( Fetch( Location->Type, "Loc_Type", "Loc_Type", 1) )

   @ 5,63 Say Location->Loc_Code Picture CAPS

   DispLatLong( Location->Latitude, 12, 26, Location->Longitude, 13, 25 )

   SysColor( "N/W" )
   KeyBoard CHR( K_ESC )
   MemoEdit(Location->Remarks, 16, 13, 20, 66, .F.)

Return .T.


* ============================================================
Function Who_Scr
* ============================================================
   Boxer( 3, 2, 23, 77, 2,,, "GR+/B" ,, .F. )

   SysColor( AACLR_BOX )
   @11, 2 Say "├"
   @11,77 Say "┤"
   @16, 2 Say "├"
   @16,77 Say "┤"

   @ 5, 6 Say "Who No:                          No. of Agency Activities:"
   @ 6, 6 Say "Agency:                                          Phone:"
```

```
   @ 7, 5 Say "Address:                                    Number of       FAX:"
   @ 8, 8 Say "City:"
   @ 8,40 Say "Activity Areas: GOV"
   @ 8,66 Say  SubStr(Defaults->PartisanHd, 1, 8)
   @ 9, 3 Say  Replicate( " ", 9 - ;
               Len(Trim(SubStr(Defaults->District, 1, 9 )))) + ;
               Proper(Trim(SubStr(Defaults->District, 1, 9))) + ":"
   @ 9,44 Say "Number of Staff:"
   @10, 3 Say  Replicate( " ", 9 - ;
               Len(Trim(SubStr(Defaults->Province, 1, 9 )))) + ;
               Proper(Trim(SubStr(Defaults->Province, 1, 9))) + ":"
   @10,42 Say "Fiscal Year Start:"

   @11, 3 Say
"-------------------------------------------------------------------------------"
   @13,25 Say "Contact:"
   @14,19 Say "Contact Title:"
   @16, 3 Say "- Remarks
-------------------------------------------------------------------------------"

   SysColor( AACLR_HEADER )
   @ 3,18 Clear To  3,61
   @ 3,19 Say "W H O 'S   D O I N G   W H A T   W H E R E"

Return .T.


* ============================================================
Function Who_Key
* ============================================================
MemVar mPermission, mWho_No

Local lResult := .T.
Local GetList := {}

   SysColor(2)     // Gets

   If mPermission <> "EXPERT"
      Return .F.
   EndIf

   If M->Is_Append
      mWho_No    := " new "
   EndIf

   @ 5,14 Say mWho_No
   SysColor(1)          // Normal

Return lResult


* ============================================================
Function Who_Fld
* ============================================================
MemVar mTag, mDistrict, mRemarks, mFY_Start, mNo_Staff, mLocation
MemVar mGov_Area, mPart_Area, mWho_No, mPermission, mAgency

Local lResult    := .F.
Local lMemo      := .F.
Local cProvince  := Space( 20 )
Local Flds_2     := { "Location", "District", "Province" }, ; // Location array
      Loc_Header := { "Location",  Proper(Trim(Defaults->District)), ;
                      Proper(Trim(Defaults->Province)) }, ; // Location header
      Flds_3     := { "Agency", "Proper(AgencyLong)" }, ;     // Agency array
      Age_Header := { "Agency:(Short Name)", "Agency:(Long Name)" }
Local Flds_5     := { "District", "Province" }, ; // District array
```

```
      Dis_Header := ( Proper(Trim(Defaults->District)), ;
                      Proper(Trim(Defaults->Province)) ) // District header
Local LocFilter  := "Location->Location != ' ' "

Local GetList    := {}

   SysColor(2)     // Gets

   If mPermission <> "EXPERT"
      Return .F.
   EndIf

   mTag := .T.              // TAG true If edit or add

   If M->Is_Append
      mDistrict  := Space( 20 )
      mRemarks   := ""
      mFY_Start  := CTOD( "01/01/SubStr(Year(Date()))" )
      mNo_Staff  := 0
   Else
   EndIf

   @ 6,14 Get mAgency BOX B_SINGLE;
            Picture CAPS ;
            Valid Help_Fill( @mAgency, "Agency", Flds_3 ;
                     ,,,,,,, Age_Header,1,, 1, 1 )

   @ 7,14 Get mAddress Picture CAPS
   @ 8,14 Get mLocation BOX B_SINGLE;
            Picture CAPS ;
            Valid Help_Fill( @mLocation, "Location", Flds_2 ;
                     ,, LocFilter,,,,, Loc_Header, ;
                     { || GetOrder("Location") } ;
                     ,, 1, 1 ) .and. ;
                        ChkDupWho( mAgency, mLocation ) .and. ;
          Pro_Loc( mLocation, @mDistrict, 09, 14, cProvince, 10, 14 )

   @ 9,14 Get mDistrict BOX B_SINGLE;
            Picture CAPS ;
            When Empty( mLocation ) ;
            Valid Help_Fill( @mDistrict, "Location", Flds_5 ;
                     ,,,,,,, Dis_Header, ;
                     { || GetOrder("District") } ;
                     ,, 1, 2 ) .and. ;
          Pro_Dist( mDistrict, 10, 14 ) // paint Province at 10,14

   @ 6,61 Get mPhone
   @ 7,61 Get mFax

   @08,61 CheckBox mGov_Area
   @08,73 CheckBox mPart_Area

   @ 9,61 Get mNo_Staff Picture [999]
   @10,61 Get mFY_Start Picture "@D"

   @13,34 Get mContact Picture CAPS
   @14,34 Get mCon_Title Picture CAPS

   lResult := Read_It( ,,GetList )

   If lResult
      If M->Is_Append
        @ 5,14 Say ( mWho_No := Next( "Who_No" )) Picture "@9"
      EndIf
   EndIf
```

```
Return lResult


* ============================================================
Function Who_Say
* ============================================================
Local lMemo    := .F.
Local nWho_Rec, nWhat_Rec
Local cWho_No := Who->Who_No
Local nActs    := 0
Local GetList := {}

   SysColor( 2 )                                    // hilighted video
   * Set Keys for tagging and untagging
   Set Key K_ALT_A To Tag_Record          // Menu to Tag or Untag all
   Set Key K_ALT_T To Tag_Record          // Tag a record
   Set Key K_ALT_U To Tag_Record                  // Untag a record
   Set Key K_ALT_X To Inn_Exit            // Exit from program

   nWho_Rec := RecNo()
   Select What
   nWhat_Rec := RecNo()                    // save record number
   Count To nActs While What->Who_No = cWho_No
   Go nWhat_Rec
   Select Who
   Go nWho_Rec

   SysColor ( "N/W" )
   @ 5,14 Say Transform( Val( Who->Who_No), "@Z" )
   @ 5,61 Say Str( nActs, 3, 0)
   @ 6,14 Say Who->Agency
   @ 7,14 Say Who->Address
   @ 8,14 Say Proper( Who->Location )
   @ 9,14 Say Proper( Who->District )
   @10,14 Say Proper( Fetch( Who->District, "Location", "Province", 2 ) )
   @08,61 CheckBox Who->Gov_Area
   @08,73 CheckBox Who->Part_Area

   @ 6,61 Say Who->Phone
   @ 7,61 Say Who->Fax

   @ 9,61 Say WHO->No_Staff
   @10,61 Say WHO->FY_Start

   @13,34 Say Proper( Who->Contact )
   @14,34 Say Proper( Who->Con_Title )

   KeyBoard CHR( K_ESC )
   MemoEdit(Who->Remarks, 17, 05, 22, 74, .F.)

Return .T.


* ============================================================
Function ChkDupWho( cAgency, cLocation )
* ============================================================
Local lResult := .F.
   If M->Is_Append
      If Seek_Rel( cAgency+cLocation, "Who", 2 )
        Pop_Msg( Trim(cAgency) + ", " + Trim(cLocation) + ;
                 " is a duplicate entry" )
      Else
        lResult := .T.
      EndIf
   Else
      lResult := T
```


B2

```
   EndIf
Return lResult


* ═══════════════════════════════════════════════════════════════════════
Function Uha_Scr
* ═══════════════════════════════════════════════════════════════════════
   Local cCur_Clr                  // When we zoom to a child table
   Boxer( 3, 2, 23, 77, 2,,, "GR+/B" ,, .F. )

   SysColor( AACLR_BOX )
   @11, 2 Say "╟"
   @11,77 Say "╢"
   @17, 2 Say "╟"
   @17,77 Say "╢"
   @19, 2 Say "╟"
   @19,77 Say "╢"

   @ 5, 6 Say "Who No:                          No. of Agency Activities:"
   @ 6, 6 Say "Agency:                                            Phone:"
   @ 7, 5 Say "Address:                               Number of       FAX:"
   @ 8, 8 Say "City:"
   @ 8,40 Say "Activity Areas: GOV"
   @ 8,66 Say  SubStr(Defaults->PartisanHd, 1, 8)
   @ 9, 3 Say  Replicate( " ", 9 - ;
               Len(Trim(SubStr(Defaults->District, 1, 9 )))) + ;
               Proper(Trim(SubStr(Defaults->District, 1, 9))) + ":"
   @ 9,44 Say "Number of Staff:"
   @10, 3 Say  Replicate( " ", 9 - ;
               Len(Trim(SubStr(Defaults->Province, 1, 9 )))) + ;
               Proper(Trim(SubStr(Defaults->Province, 1, 9))) + ":"
   @10,42 Say "Fiscal Year Start:"
   @11, 3 Say
"═══════════════════════════════════════════════════════════════════════"
@12, 8 Say "File No:"
@13, 1 Say "Activity No:              Title:"
@14, 4 Say "Sector Name:"
@15, 7 Say "Location:                                      Starting Date:"
@16, 5 Say  Replicate( " ",10 - ;
           Len(Trim(SubStr(Defaults->District, 1,10 )))) + ;
           Proper(Trim(SubStr(Defaults->District, 1,10))) + ":"
@16,55 Say "Ending Date:"
@17, 3 Say "──────────── ──────────────────────────────────────────────────"
@18, 4 Say "Description:"
@19, 3 Say "──────────────────────────── Remarks ──────────────────────────"

   SysColor( AACLR_HEADER )
   @ 3,18 Clear To  3,61
   @ 3,19 Say "W H O 'S   D O I N G   W H A T   W H E R E"
   @11,32 Clear To 11,48
   @11,33 Say "A C T I V I T Y"

   If EOF()                            // THE_USUAL() sets this .T.
      cCur_Clr := SysColor()           // which has no matching record
      SysColor( 2 )                    // Get color
      @ 5,14 Say Transform( Val( Who->Who_No), "@Z" )
      @ 6,14 Say Who->Agency
      @ 7,14 Say Proper( Who->Address )
      @ 8,14 Say Proper( Who->Location )
      @ 6,61 Say Who->Phone
      @ 7,61 Say Who->Fax
      @ 9,61 Say Who->No_Staff
      @10,61 Say Who->FY_Start
      SetColor( cCur_Clr )
   EndIf
```

```
Return .T.


* ============================================================
Function Wha_Key
* ============================================================
MemVar mPermission, mWhat_No

Local lResult := .T.
Local GetList := {}

   SysColor(2)     // Gets

   If mPermission <> "EXPERT"
      Return .F.
   EndIf

   If M->Is_Append
      mWhat_No    := " new "
   EndIF

   @ 5,14 Say Transform( Val( Who->Who_No), "@Z" )
   @ 6,14 Say Who->Agency
   @ 7,14 Say Proper( Who->Address )
   @ 8,14 Say Proper( Who->Location )
   @ 6,61 Say Who->Phone
   @ 7,61 Say Who->Fax
   @ 9,61 Say Who->No_Staff
   @10,61 Say Who->FY_Start

   @12,17 Say Transform( Val( mWhat_No), "@Z" )
   SysColor(1)          // Normal

Return lResult


* ============================================================
Function Wha_Fld
* ============================================================
MemVar mPermission, mTag, mWho_No, mRemarks, mActNo, mLocation, mDistrict
MemVar mWhat_No

Local lResult    := .F.
Local lMemo      := .F.
Local cDistrict  := Space( 20 )
Local cProvince  := Space( 20 )
Local Flds_1     := { "ActNo", "Proper(Title)" },;       // Activity array
      Act_Header := { "Activity:Number", "Activity:Title" }  // Activity header
Local Flds_2     := { "Location", "Proper( District )", "Proper( Province )", ;
            "Proper( TYPE )" }, ; // Location array
      Loc_Header := { "Location",  Proper(Trim(Defaults->District)), ;
            Proper(Trim(Defaults->Province)), "Type" }    // Location header
Local Flds_5     := { "District", "Province" }, ; // District array
      Dis_Header := {  Proper(Trim(Defaults->District)), ;
                  Proper(Trim(Defaults->Province)) } // District header
Local LocFilter  := "Location->Location != ' ' "
LocaL ActFilter  := "Len( Trim( ActTitle->ActNo ) ) > 2"

Local GetList    := {}

   SysColor(2)     // Gets

   If mPermission <> "EXPERT"
      Return .F.
   EndIf
```

```
   mTag := .T.                  // TAG true If edit or add

   If M->Is_Append
      mWho_No     := Who->Who_No
      mRemarks    := ""
   Else
      mWho_No     := What->Who_No
   EndIf

   @13,17 Get mActNo BOX B_SINGLE;
              Picture "@K 9.9.99" ;
              Valid Help_Fill( @mActNo, "ActTitle", ;
                Flds_1 ,, ActFilter,,,,, Act_Header,,, 1, 1 ) // .and. ;
                Lookup( mActNo, "ActTitle", "Proper(Title)",13, 36 )

   @14,17 Say SubStr( Proper( Fetch( SubStr(mActNo, 1, 3) + "   ", ;
                    "ActTitle", "Title", 1 ) ), 1, 40 )

   @15,17 Get mLocation ;       // BOX B_SINGLE;
              Picture CAPS ;
              Valid ( Empty( mLocation ) .or. ;
                   Help_Fill( @mLocation, "Location", Flds_2 ;
                        ,, LocFilter,,,,, Loc_Header, ;
                        ( || GetOrder("Location") ) ;
                        ,, 1, 1 ) .and. ;
                        ChkDupWhat( mActNo, mLocation, mDistrict ) .and. ;
             Pro_Loc( mLocation, @mDistrict, 16, 17, cProvince, 16, 39 ) ) ;
               HELP "Any key + <ENTER> for picklist"

   @16,17 Get mDistrict BOX B_SINGLE;
              Picture CAPS ;
              When Empty( mLocation ) ;
              Valid  Help_Fill( @mDistrict, "Location", Flds_5 ;
                        ,,,,,,, Dis_Header, ;
                        ( || GetOrder("District") ) ;
                        ,, 1, 2 ) .and. ;
                            ChkDupWhat( mActNo, mLocation, mDistrict ) .and. ;
            Pro_Dist( mDistrict, 17, 17 ) // paint Province at 16,39

   @15,68 Get mStart_Date Picture "@D"
   @16,68 Get mEnd_Date Picture "@D"

   @18,17 Get mDescrip Picture CAPS

   lResult := Read_It( ,,GetList )

   If lResult
      If M->Is_Append
      @12,17 Say ( mWhat_No := Next( "What_No" )) Picture "@9"
      EndIf
   EndIf

Return lResult


* ============================================================
Function Wha_Say
* ============================================================
Local lMemo    := .F.
Local cCur_Clr := SysColor()
Local nWhat_Rec, nWho_Rec
Local cWho_No  := Who->Who_No
Local nActs    := 0
Local GetList  := {}

   SysColor( 2 )                                  // hilighted video
```

```
   * Set Keys for tagging and untagging
   Set Key K_ALT_A To Tag_Record          // Menu to Tag or Untag all
   Set Key K_ALT_T To Tag_Record          // Tag a record
   Set Key K_ALT_U To Tag_Record                 // Untag a record
   Set Key K_ALT_X To Inm_Exit            // Exit from program

   nWhat_Rec := RECNO()                    // save record number
   Select Who
   nWho_Rec  := RecNo()
   Select What
   Go Top
   If Seek_Rel( cWho_No, "What" )
      Count To nActs While What->Who_No = cWho_No
   Else
      Pop_Msg( "Unable to find " + cWho_No + " in WHAT File" )
   EndIf
   Select Who
   GO nWho_Rec
   Select What
   GO nWhat_Rec

   SysColor( 2 )                          // Get color
   @ 5,14 Say Transform( Val( cWho_No), "@Z" )
   @ 5,61 Say Str( nActs, 3, 0)
   @ 6,14 Say Who->Agency
   @ 7,14 Say Who->Address
   @ 8,14 Say Proper ( Who->Location )
   @ 9,14 Say Proper( Who->District )
   @10,14 Say Proper( Fetch( Who->District, "Location", "Province", 2 ) )
   @08,61 CheckBox Who->Gov_Area
   @08,73 CheckBox Who->Part_Area
   @ 6,61 Say Who->Phone
   @ 7,61 Say Who->Fax
   @ 9,61 Say Who->No_Staff
   @10,61 Say Who->FY_Start

   SysColor ( "N/W" )
   @12,17 Say Transform( Val( What->What_No), "@Z" )
   @13,17 Say What->ActNo
   @13,36 Say Proper( Fetch( What->ActNo, "ActTitle", "Title", 1 ) )
   @14,17 Say SubStr( Proper( Fetch( SubStr(What->ActNo, 1, 3) + "   ", ;
                  "ActTitle", "Title", 1 ) ), 1, 40)
   @15,17 Say Proper( What->Location )
   @16,17 Say Proper( What->District )
   @16,39 Say SubStr( Proper( Fetch( ;
             What->District, "Location", "Province", 2 )), 1, 15)

   @15,68 Say What->Start_Date
   @16,68 Say What->End_Date
   @18,17 Say Proper( What->Descrip )


   KeyBoard CHR( K_ESC )
   MemoEdit(What->Remarks, 20, 04, 22, 75, .F.)

Return .T.


* ============================================================
Function ChkDupWhat( cActNo, cLocation, cDistrict )
* ============================================================
Local lResult := .F.
   If M->Is_Append
      If Seek_Rel( cActNo+cLocation+cDistrict, "What", 4 )
        Pop_Msg( Trim(cActNo) + ", " + Trim(cLocation) + ", " + ;
```

```
                Trim( cDistrict ) + " is probably a duplicate entry" )
      Else
        lResult := .T
      EndIf
   Else
      lResult := .T.
   EndIf
Return lResult


* ============================================================
Function StripSoft
* ============================================================
Local lResult := .T
Local cDbf := Alias(), cFileName
Local cSaveFile   := Select()
Local cSaveRecNo  := RecNo()

   cDbf := cDbf + Space( 8 - Len( cDbf ) )
   cFileName := Ask_For( "Name of file with memos to convert", ;
                  GcDbf, "!!!!!!!!")

   cDbf := Trim( cDbf )
   Conv_Memo( cDbf )

   Select( cSaveFile )
   Go cSaveRecNo

Return lResult

* ============================================================
Function Conv_Memo( cDbf )
* ============================================================
* Get rid of soft carriage Returns to print out remarks in R&R
* replace hard carriage Returns with a hard carriage Return..
* replace soft carriage Returns with spaces.
Local lResult     := .T.
Local cSaveFile   := Select()
Local cSaveRecNo  := RecNo()
Local aMfields    := {}
Local nArrayItems := 0
Local cMemo, cTranMemo, cMemoField
Local i           := 1
Local bBlock

   Select (cDBF)                        // Get file scanned in report
   aMfields    := MemoFields()          // make array of names of memo fields
   nArrayItems := Len( aMfields )       // number of memo fields

   For i = 1 To nArrayItems             // convert each memo field
      Go Top
      cMemo      := aMfields[i]
      bBlock     := FieldBlock( cMemo )
      cMemoField := Trim(cDBF) + "->" + Trim(cMemo)
      Do While ! Eof()
        cTranMemo := StrTran( &cMemoField, Chr(141) + Chr(10))
        Replace_Em( bBlock, cTranMemo, .F. )
        DbSkip()
      EndDo
   Next
   Go Top

   Select( cSaveFile )
   Go cSaveRecNo
Return lResult
```

```
*
Function Conv_Mem
*
* Get rid of soft carriage Returns to print out remarks in R&R
* replace hard carriage Returns with a hard carriage Return..
* replace soft carriage Returns with spaces.
Local lResult      := .T.
Local cSaveFile    := Select()
Local cSaveRecNo   := RecNo()
Local aMfields     := {}
Local cDbf         := Reports->Ri_Alias
Local nArrayItems  := 0
Local cMemo, cTranMemo, cMemoField
Local i            := 1
Local bBlock

   Select (cDBF)                          // Get file scanned in report
   aMfields    := MemoFields()            // make array of names of memo fields
   nArrayItems := Len( aMfields )         // number of memo fields

   For i = 1 To nArrayItems               // convert each memo field
      Go Top
      cMemo      := aMfields[i]
      bBlock     := FieldBlock( cMemo )
      cMemoField := Trim(cDBF) + "->" + Trim(cMemo)
      Do While ! Eof()
        cTranMemo := StrTran( &cMemoField, Chr(141) + Chr(10))
        Replace_Em( bBlock, cTranMemo, .F. )
        DbSkip()
      EndDo
   Next
   Go Top

   Select( cSaveFile )
   Go cSaveRecNo
Return lResult


*
Function GetActDonors( cDonors, nRow, nCol, lUpdate, lPaint, lDonorsOnly, ;
                       nDisplayLength, nDonorFieldLen )
*
* Say Donors in Activity by making a string of the the pledge donors
* for that same Activity by constructing an array of unique donors,
* sorting them and then returning them as a string either for inclusion
* in the Activity file or to merely SAY.  lUpdate is .T. if Activity->Donor
* to be updated.  lDonorsOnly is .T. if only pledges with "Donor" checked in
* entry_type.

Local lResult        := .F.
Local cSaveFile      := Select()
Local cSaveRecNo     := RecNo()

Local nPledgeRecNo   := 0
Local nActivityRecNo := 0
Local cDonorString   := ""
Local aDonorArray    := {}
Local nArrayItems    := 0
Local cActNo         := Activity->ActNo
Local nStayOnRecord  := 2
Local bBlock
Local i              := 1

   Select Pledge
   nPledgeRecNo    := RecNo()
```

```
// check for pledge records for this activity
If Seek_Rel( cActNo, "Pledge"  )
   lResult := .T.
Else
   // no pledge records found for this activity
EndIf

If lResult   // pledge records exist so construct array of donors

   Select Pledge
   Go Top
   Seek_Rel( cActNo, "Pledge", 1,, nStayOnRecord )
   Do While Pledge->ActNo == cActNo
     // construct array of unique donors
     If aCount( Trim( Pledge->Donor ), aDonorArray ) == 0
        // this is a new donor -- include in array but check for
        // donors only first
        If lDonorsOnly
           // Donors only, so check for "D" in Pledge Entry_Type field
           If "D" $ Pledge->Entry_Type
              aAdd( aDonorArray, Trim( Pledge->Donor ) )
           EndIf
        Else
           // add new element regardless of whether record is a donor
           aAdd( aDonorArray, Trim( Pledge->Donor ) )
        EndIf
     EndIf
        // skip to the next pledge record
        DbSkip()
   EndDo

   // array is now complete -- sort it next
   aSort( aDonorArray )
   // check length of array
   nArrayItems := Len( aDonorArray )

   If nArrayItems > 1
     For i = 1 To nArrayItems
        If i == 1 .or. nArrayItems == 1
           // end of array or array of 1 element -- don't add "/"
           cDonorString := cDonorString + aDonorArray[i]
        Else
           // add "/" to string before appending new donor
           cDonorString := cDonorString + "/" + aDonorArray[i]
        EndIf
     Next
   EndIf

   If lPaint
     // paint donor string back to screen
     // if donor string greater than nDisplayLength chars,
     // chop off last 3 chars
     // and add "..." to indicate that not all the string is showing.
     @ nRow, nCol Say IIF( Len( cDonorString ) > nDisplayLength, ;
              SubStr( cDonorString, 1, nDisplayLength - 3 ) + ;
              "...", PadR( cDonorString, nDisplayLength ) )
   EndIf

   // if lUpdate, replace Pledge->Donor with new donor string
   If lUpdate
     cDonorString := IIF( Len( cDonorString ) > ;
                 nDonorFieldLen, SubStr( cDonorString, 1, ;
                 nDonorFieldLen ), cDonorString )
     Activity->( Replace_En( "Donors", cDonorString ) )
     cDonors := cDonorString
   EndIf
```

```
      // restore pledge file to original record number
      Go nPledgeRecNo

   Else
      // no pledge records found -- exit function
   EndIf

   Select( cSaveFile )
   Go cSaveRecNo
Return .T.


* ============================================================
Function UpdateActDonors( )
* ============================================================
// Update all Activity Donor records for printing reports

Local lResult        := .F.
Local cSaveFile      := Select()
Local cSaveRecNo     := RecNo()
Local nActivityRecNo

   Select Activity
   nActivityRecNo := RecNo()
   Go Top
   Do While !EOF()
      // parameters: cDonors, nRow, nCol, lUpdate, lPaint, nDisplayLength,
      //             nDonorFieldLength
      GetActDonor( Activity->Donors, 0, 0, .T., .F., 0, 126 )
      dbSkip()
   EndDo
   Go nActivityRecNo

   // cleanup and exit
   Select( cSaveFile )
   Go cSaveRecNo
Return .T.


* ============================================================
Function UpDnLoad
* ============================================================
MemVar mUserNum

Local cOld_Help := HelpCode( ProcName() )

Local Options := { ;
                   "UpLoad selected files  ",  ;
                   "DownLoad selected files"   ;
                 },;
  Permissions := { ;
                   .T., ;
                   IIF( mUserNum == "1   " .or. mUserNum == "2   ", .T., .F.) ;
                 },;
    Decisions := { ;
                   "UpLoad()", ;
                   "DnLoad()"  ;
                 }

   Pull_Open( Options, 3, 55, 1, Permissions, Decisions)
   HelpCode( cOld_Help )
Return .T.


* ============================================================
Function UpLoad
```

```
*
Field Tag
Local cOld_Help := HelpCode( ProcName() ), nOldOrder := SetOrder()

Local files_upld := { ;
    "Activity", ;
    "Acttitle", ;
    "Address", ;
    "Agency", ;
    "Category", ;
    "Donor", ;
    "Incident", ;
    "Location", ;
    "Mail", ;
    "Mines", ;
    "Mine_ass", ;
    "Needs", ;
    "Pledge", ;
    "Populate", ;
    "Partisan", ;
    "Resource", ;
    "What", ;
    "Who"  ;
   }

Local No_Of_Files := Len(Files_Upld)
Local Total_Space := 0
Local No_Tagged   := 0
Local Tagged[No_Of_Files]
Local Last_Char
Local i
Local cTempScrn   := SaveScreen(23, 2, 23, 32)
Local lResult     := .T.
Local msg2        := ""
Local file_name   := ""
Local cpath       := 'C:\Clipper5\Artful\SHAMAN\UpDnLoad'
Local ndrive      := 0
Local drivespace  := 0
Local nNumRec     := 0
Local Target_Dru  := ""

   ASort(Files_Upld)
   AFill(Tagged,.T.)

   StatusLine("SPACEBAR=tag on/off, ESC=quit, CTRL-T=tag all, " + ;
              "CTRL-U=untag all, ENTER=accept")
   No_Tagged := T_Menu(Files_UpLd, Tagged,,,,,, "UpLoad")

   * If no files tagged, Return
   If No_Tagged != 0

      * Compute disk space necessary to hold upload files
      FOR i := 1 TO no_of_files
        If Tagged[i]
           Select (Files_UpLd[i])
           Count To nNumRec For Tag
           * Size is approximate allowing 500 bytes for a memo for each
           * record.
           Total_Space := Total_Space + (nNumRec * 500) + ;
              ((nNumRec * RecSize()) + Header() +1 )
           StatusLine( Files_Upld[i] + ' File size : '+ ;
              LTrim(Str( ((nNumRec * RecSize()) + Header() +1 ))) + ;
              " bytes.  " + "Total needed: " + LTrim(Str(Total_Space)))
        EndIf
      Next
```

```
  Target_Drv  := Pad(cPath, 35)
  Ask_For("Enter UpLoad Path", @Target_Drv, "@!" )

  Last_Char := IF(Right(Target_Drv,1) = '\', "", '\')
  Do While !Is_Dir(Target_Drv + Last_Char)
    * Whoops, path does not exist - let user try again if desired
    If Verify(Target_Drv + " path does not exist - retry ?")
       Target_Drv  := Pad(Target_Drv, 35)
       Ask_For("Enter UpLoad Path", @Target_Drv, "@!" )
    Else
       Return .F.
    EndIf
  ENDDO

  * Whoops, path exists, but it is the same as data path
  * let user try again if desired
  Do While Target_Drv == CHR(CurDriv()) + ':\' + CurDir()
    If Verify(Target_Drv + " same as data path - retry ?")
       Target_Drv  := Pad(Target_Drv, 35)
       Ask_For("Enter UpLoad Path", @Target_Drv, "@!" )
       * Whoops, path does not exist - let user try again if desired
       Last_Char := IF(Right(Target_Drv,1) = '\', "", '\')
       Do While !Is_Dir(Target_Drv + Last_Char)
          If Verify(Target_Drv + " path does not exist - retry ?")
            Target_Drv  := Pad(Target_Drv, 35)
            Ask_For("Enter UpLoad Path", @Target_Drv, "@!" )
          Else
            Return .F.
          EndIf
       EndDo
    Else
       Return .F.
    EndIf
  EndDo

  * Everything OK - see If user wants to continue
  If ! Verify( "UpLoad to " + Target_Drv + " ?" ) .AND. No_Tagged > 0
    Return lResult
  EndIf

  * Compare actual diskspace with what is needed
  * First convert drive letter (A to H) to a number from 1 to 8
  FOR i := 1 TO 8
    If ASC(Target_Drv) = 64 + i
       nDrive := i
       Exit
    EndIf
  Next

  DriveSpace := DiskSpace(nDrive)
  Msg2 := LTrim(Str(DriveSpace)) + " on disk, but " + ;
    LTrim(Str(Total_Space)) + " required."

  * If not enough disk space, pop a message and Return
  If Total_Space >= DriveSpace
    Pop_Msg(("Error .. not enough space on disk " + Left(Target_Drv,1), ;
       Msg2, ;
       "Change drive or delete some files and try again."})
  Else
    * OK, we got enough space, now Get on with it.
    For i := 1 To No_Of_Files
       File_Name := Files_Upld[i]
       If Tagged[i]
          StatusLine ( 'Copying file : '+ Files_Upld[i])
          Select (File_Name)
          Last_Char := IF(Right(Target_Drv,1) = '\', "", '\')
```

```
            Copy All To (Target_Dru + Last_Char + File_Name) For Tag
          EndIf
       Next
       StatusLine ( 'File UpLoad(s) completed to ' + Target_Dru)
     EndIf
  EndIf

  If Verify( "View Uploaded Files ?" )
     ShowDir( Target_Dru )
  EndIf

  RestScreen(23, 2, 23  32, cTempScrn)

  HelpCode( cOld_Help )
  SetOrder ( nOldOrder )
Return lResult


* ============================================================
Function DnLoad
* ============================================================
Local cOld_Help := HelpCode( ProcName() )
Local cDrive       := CHR( CurDriv() )
Local cTargetPath  := cDrive + ":\" + CURDIR( cDrive )
Local cSourcePath  := PAD('A:\', 35)

Local aFile        := {}
Local aFilesDnld   := {}
Local aFilesNTX    := {}
Local nNoFiles     := 0
Local nNoTagged    := 0
Local aTagged      :={}
Local aFlds        :={}
Local i, j, k
Local cTempScrn    := SaveScreen( 8, 25, 21, 59 )
Local lResult      := .T.
Local cOldHelp     := HelpCode( ProcName() )
Local cTempFile    := ""
Local cTFile       := ""
Local nAppend      := 0
Local nFound       := 0

Local cFile        := ""
Local nFlds        := 0
Local cDnFile      := ""
Local cTargetNtx   := ""
Local cTarget      := ""
Local cDnFileNtx   := ""
Local cTemp, Temp3, Temp1
Local mIs_Append
Local cRepSource, cRepTarget

  SysColor( AACLR_NORMAL )

  Ask_for("Enter Source Path",@cSourcePath,"@!")
  cSourcePath := Trim(cSourcePath)
  nNoFiles := ADir( cSourcePath+"*.DBF" )
  If nNoFiles = 0
     Pop_Msg( "No Database Files found on " + cSourcePath + " - Quitting..." )
     Return .F.
  EndIf
  If Verify( "View Source Files ?" )
     ShowDir(cSourcePath)
  EndIf

  Ask_for("Enter Target Path",@cTargetPath,"@!")
```

```
   ASize( aFilesDnld, nNoFiles )
   nNoFiles := Adir( cSourcePath+"*.DBF",aFilesDnld )
   Asize( aTagged, nNoFiles )
   Afill( aTagged, .T. )
   Asize( aFilesNtx, nNoFiles )

   If ! Verify( "Continue Download ?" )
      RestScreen( 8, 25, 21, 59, cTempScrn)
      Return .F.
   EndIf

   SysColor( "GR+/BR" )
   Boxer( 8, 25, 20, 58, 3,, " ", "GR+/BR" ,, .T. )
   @14,25 Say "├"
   @14,58 Say "┤"
   @10,28 Say "Downloading    of    Files"
   @12,28 Say "File Name:  "
   @14,26 Say "──────────────────────────────"
   @15,38 Say "RECORDS"
   @17,34 Say "Replaced:  "
   @18,34 Say "Appended:  "

   SysColor(AACLR_HEADER)
   @ 8,34 Clear To  8,50
   @ 8,35 Say "D O W N L O A D"
   @ 10,46 Say nNoFiles Picture [99]

   For i := 1 To nNoFiles
      cTempFile := SubStr( aFilesDnld[i], 1, Len(aFilesDnld[i]) - 4 )
      If Seek_Rel( cTempFile, "DICT" )
        aFilesNTX[ i ] := DICT->Key_1
      Else
        Pop_Msg(( "Illegal file "+aFilesDnld[i]+" found on DownLoad Disk", ;
                  "                          Continuing..." ))
      EndIf
   Next

   StatusLine("SPACEBAR=tag on/off, ESC=quit, CTRL-T=tag all, CTRL-U=untag all,
ENTER=accept")

   nNoTagged := T_Menu( aFilesDnLd, aTagged,,,,,, "DownLoad" )

   If ! Verify( "Continue Download ?" )
      RestScreen( 8, 25, 21, 59, cTempScrn)
      Return .F.
   EndIf

   // If no files tagged, Return
   If nNoTagged != 0
      For i := 1 To nNoFiles
        If ! aTagged[i] .or. aFilesNTX[i] = NIL
           Loop
        Else
           // Main work takes place here
           cFile     := cSourcePath + aFilesDnld[i]
           cTempFile := SubStr( aFilesDnld[i], 1, Len(aFilesDnld[i]) - 4 )
           Select 0
           Use &cFile Alias "SOURCE"
           cRepSource := SubStr( cFile, 1, Len(cFile) - 4)
           aFlds := DbfFields()
           nFlds := Len( aFlds )
           cRepTarget := SubStr( cDnFile, 1, Len(cDnFile) - 4)
           cDnFile    := "c:\clipper5\artful\shaman\" + aFilesDnld[i]
           cDnFileNtx := IIF( Len(cTempFile) < 8, cTempFile + "1", ;
                         SubStr(cTempFile,1,7) + "1" )
           cDnFileNtx := "c:\clipper5\artful\shaman\" + cDnFileNtx
```

```
            Select 0
            Use &cDnFile index &cDnFileNtx alias "TARGET"

            @ 12,39 Say PADC(cTempFile, 8, " ")
            cTargetNtx := aFilesNTX[i]
            Select Source

            Do While ! EOF()
               cTarget    := &cTargetNtx
               Select Target
               Seek cTarget
               If ! Found()
                 // Append a blank and replace fields
                 AddRec()
                 nAppend++
               Else
                 nFound++
               EndIf
               // Replace Fields
               For j := 1 To nFlds
                 k := j
                 cTemp := aFlds[k]
                 temp3 := SOURCE->( &cTemp )
                 temp1 := TARGET->( cTemp )

                 Replace &temp1 With temp3
               Next

               Target->TAG := .F.

               Select Source
               Skip
               @ 17, 44 Say nFound Picture [9999]
               @ 18, 44 Say nAppend Picture [9999]
            EndDo
            // Prepare for next download file
            Close Source

            Close Target
            nFound  := 0
            nAppend := 0
         EndIf
      Next
   EndIf
   Pop_Msg( { "       DownLoad Completed", ;
            " Will now reindex ALL existing files" } )

    // Erase file's indexes
    Close All
*    AEval(Directory("*.ntx"), { |aFile| FErase(aFile[F_Name]) } ) && Why it's
                                                          && not working ?

   If !chk_dict( DictFile,,.T. )   // open the dictionary and force reindex..
      Set Cursor On
      Quit
   EndIf

   RestScreen( 8, 25, 21, 59, cTempScrn)

   HelpCode( cOld_Help )
Return lResult

*  ============================================================
Function File_Size( Tagged_File )
*  ============================================================
Select (Tagged_File)
```

```
Return ((LastRec() * RecSize()) + Header() +1 )


* ============================================================ *
FUNCTION FileSize( cFileName )
* ============================================================

Local nLength := 0
Local nHandle := 0
   // Open the file read-only
   IF( nHandle := FOpen( cFileName )) >= 0
      // Get length of file
      nLength := FSeek( nHandle, 0, FS_END )
      // Reset file position to beginning of file
      FSeek( nHandle, 0 )
      FClose( nHandle )
   Else
      Pop_Msg( "File open error: " + Str( FError() ) )
   EndIf
Return nLength


* ============================================================
External dbEdit, ADir, ASort                    // Macro calls and overlaid funcs
*                                               // must be declared External.

External Get_Std2, Dust_1, m2ap, Replace_Em
External PackFile, PackDict, U_Edit, U_tBrowse, UGet, U_Recedit, ;
   The_Usual, U_Browse, U_Calc, U_Memo, U_Query, U_Search, U_Zoom

External Init_Query, Is_Query, QueryName, Push_Query, Pop_Query

External Do_Swap, Mask_For, FldInkey, Set_Relat, Lookups, Is_Lookup, ;
   Check_Rel, Keep_Relat, GetLookups, SetLookups, ;
   Push_Rel, Pop_Rel, Children, Parents, Search, Set_Table, U_Search

External Exist, Get_File, Key_Value, Make_Table, Next, PathFind, Unique
External Help, ShowDir, HelpCode

External Dct_Maint, Dct_Scr, Dct_Say, Dct_Key, Dct_Fld, ;
   Chk_Dict, DictFind, DictOpen, db_Open

External Rpl, RplChiFlds, Make_Table, Set_Prn, ;
   List_Em, K_Scroll, Auto_Init, ;
   Count_Em, Sum_Fields, Query, ;
   Menu_Ord, xCount, Sign_On, Help_Fill, ;
   TextDate, TextTime, ViewFile, ScrollFile, ;
   ScrapFiles, Tag_Record, Help_Ntx, Help_xCpt, Hlp_Edit

External Sys_Scr, Uqr_Scr

External Repo_Man                               // Not needed to run reports, but
// useful developing R&R reports.
External Rpt_Menu, Rpt_Exec                     // Required for all reports
External Rpt_RRun                               // RRUNTIME caller
External Rpt_Std                                // FRM/LBL reports
External Rpt_Call, Rpt_Dest, Rpt_Opts           // reports
External Rpt_Scope, Rpt_Query, Rpt_Headr

External xBrowser, xB_Udf, Set_Table, LogIn
External QuitProg, Pop_Msg

External Seek_Rel
External MemoTran, ColorInit, Aud_Before
```

```
*  ╔═══════════════╦═══════════════════════════════════════════════════════╗
*  ║ Program       ║ SHA_MAS.PRG                                           ║
*  ║ System        ║ ARTFUL.LIB for Clipper 5.01                           ║
*  ║ Functions     ║ Mas_Scr, Mas_Key, Mas_Fld, Mas_Say, Pro_Dist          ║
*  ║               ║                                                       ║
*  ║ Purpose       ║ Maintain an audit trail of all changes                ║
*  ║ Conventions   ║ None.                                                 ║
*  ║ Syntax        ║                                                       ║
*  ║ Parameters    ║ None                                                  ║
*  ║ Example       ║                                                       ║
*  ║ Assumes       ║                                                       ║
*  ║ Side Effects  ║                                                       ║
*  ║ Returns       ║                                                       ║
*  ║               ║                                                       ║
*  ║ ARTFUL Calls  ║ THE_USUAL, BOXER, SYSCOLOR, UGET, READ_IT             ║
*  ║ Authors       ║ David W. Zimmerly                                     ║
*  ║ Copyright     ║ BlueWater Software 1991- 1993                         ║
*  ╚═══════════════╩═══════════════════════════════════════════════════════╝

#include "artful.ch"
#include "box.ch"
#include "set.ch"
#include "aa_pulld.ch"
#include "aa_gets.ch"

#xtranslate M( <foo> )     -> aRecord\[ FieldPos( <"foo"> ) \]
#xtranslate MStr( <foo> ) -> aRecord\[ FieldPos( <foo> ) \]

#define CAPS       '@K!'
#define cont_pic   '@K! 9999999XXXXXXXXXXX'
#define AACLR_BOX "GR+/B"

*  ═══════════════════════════════════════════════════════════════════════
FUNCTION MAS_SCR
*  ═══════════════════════════════════════════════════════════════════════
Local lResult := .T.
   Local SubHeader := "GR+/B"

   SysColor( AACLR_BOX )
   BOXER( 2, 0, 23, 79, 2,,, "GR+/B" ,, .F. )
   @ 5, 5 Say "Number:                                        Location:"
   @ 6, 7 Say "Date:"
   @ 6,42 Say Replicate( " ", 10 - ;
              Len(Trim(SubStr(Defaults->District, 1,10)))) + ;
              Proper(Trim(SubStr(Defaults->District, 1, 10 ))) + ":"
   @ 7, 3 Say "Reporter:"
   @ 7,42 Say Replicate( " ", 10 - ;
              Len(Trim(SubStr(Defaults->Province, 1,10)))) + ;
              Proper(Trim(SubStr(Defaults->Province, 1, 10))) + ":"
   @ 9, 3 Say "Related                                        Map Sheet:"
   @10, 3 Say "Details                                         Grid Ref:"
   @11, 3 Say "Details                                         Latitude:"
   @12,43 Say "Longitude:"
   @15,47 Say "Types:"
   @17,45 Say "Numbers:"
   @19,45 Say "Laid By:"
   @21,43 Say "Date Laid:"

   SysColor( AACLR_HEADER )
   @ 2,25 CLEAR TO  2,55
   @ 2,26 Say "M I N E   A S S E S S M E N T"

   BOXER( 4, 13, 8, 41, 1,,, "W+/B" ,, .F. )
   SysColor( AACLR_HEADER )
   @ 4,15 CLEAR TO  4,22
   @ 4,16 Say "REPORT"
```

```
   BOXER( 4, 54, 13, 77, 1,,, "W+/B" ,, .F. )
   SysColor( "W+/B" )
   @ 8,54 Say "├"
   @ 8,77 Say "┤"
   @ 8,55 Say "─                    ─────────"

   SysColor( AACLR_HEADER )
   @ 4,56 CLEAR TO  4,65
   @ 4,57 Say "LOCATION"
   @ 8,56 CLEAR TO  8,68
   @ 8,57 Say "COORDINATES"

   BOXER( 9, 13, 13, 41, 1,,, "W+/B" ,, .F. )
   SysColor( AACLR_HEADER )
   @ 9,15 CLEAR TO  9,34
   @ 9,16 Say "INFORMATION SOURCE"

   BOXER( 11, 3, 13, 9, 1,,, "W+/B" ,, .F. )

   BOXER( 14, 54, 22, 77, 4,,, "W+/B" ,, .F. )
   SysColor( "W+/B" )
   @16,54 Say "╟"
   @16,77 Say "╢"
   @18,54 Say "╟"
   @18,77 Say "╢"
   @20,54 Say "╟"
   @20,77 Say "╢"
   @16,55 Say "──────────────────────"
   @18,55 Say "──────────────────────"
   @20,55 Say "──────────────────────"

   SysColor( AACLR_HEADER )
   @14,56 CLEAR TO 14,69
   @14,57 Say "MINE DETAILS"

   BOXER( 14, 03, 22, 41, 4,,, "W+/B" ,, .F. )
   SysColor( "W+/B" )
   @17, 3 Say "╟"
   @17,41 Say "╢"
   @20, 3 Say "╟"
   @20,41 Say "╢"
   @17, 4 Say "──────────────────────────────────────"
   @20, 4 Say "──────────────────────────────────────"
   @14, 5 CLEAR TO 14,20
   @14, 6 Say "REPORTED MINES"

   SysColor( AACLR_HEADER )
   @14,05 CLEAR TO 14,20
   @14,06 Say "REPORTED MINES"
   @17,05 CLEAR TO 17,21
   @17,06 Say "SUSPECTED MINES"
   @20,05 CLEAR TO 20,13
   @20,06 Say "REMARKS"

Return lResult


* ═══════════════════════════════════════════════════════════
FUNCTION MAS_KEY
* ═══════════════════════════════════════════════════════════
MemVar mPermission, mTag, mReport_No, mRep_Date
Local lResult := .F.

Local GetList := {}
```

```
   SysColor(2)     // Gets

   If mPermission <> "EXPERT"
      RETURN  F
   EndIf

   mTag := T               // TAG true If edit or add

   If M->Is_Append
      mReport_No  := "new "
      mRep_Date   := Date()
   Else
   EndIF

   @ 5,15 Say mREPORT_NO
   @ 6,15 Get mREP_DATE PICTURE "@D" ;
                    VALID mRep_Date > Ctod( "01/10/92" ) .and. ;
                          mRep_Date <= Date()

   IF lResult := Read_It( ,,getList )
      If M->Is_Append
      Else
      EndIf
   EndIf

   SysColor(1)          // Normal

Return lResult


* ============================================================
FUNCTION MAS_FLD
* ============================================================
MemVar mDistrict, mLatitude, mLongitude, mSource, mReported
MemVar mNo_Mines, mSuspected, mRemarks, mPermission
MemVar mLocation, mTag, mReport_No

Local lResult := .F.

Local flds_2     := { "LOCATION", "DISTRICT", "PROVINCE" }, ; // Location array
      Loc_Header := { "Location", "District", "Province" }    // Location header

Local flds_5     := { "DISTRICT", "PROVINCE" }, ; // District array
      Dis_Header := { "District", "Province" } // District header

Local cProvince := Space( 20 )
Local lMemo     := .F.
Local LocFilter  := "Location->Location != ' ' "
Local cNS_Lat     := "S", ;
      cEW_Long    := "E"

Local GetList := {}

   SysColor(2)     // Gets

   If M->Is_Append
      mDistrict  := Space( 20 )
      mLatitude  := 0.0
      mLongitude := 0.0
      mSource    := ""
      mReported  := ""
      mNO_MINES  := 1
      mSuspected := ""
      mRemarks   := ""
      cNS_Lat    := "S"
      cEW_Lorg     "E"
```

```
Else
   mLatitude  := LatDeg( Mine_Ass->Latitude, @cNS_Lat )
   mLongitude := LongDeg( Mine_Ass->Longitude, @cEW_Long )
Endif

If nPermission <> "EXPERT"
   RETURN .F.
EndIf


SYSCOLOR ( ",N/W" )

@ 7,15 Get mREP_NAME PICTURE CAPS

@ 5,56 Get mLOCATION ;    // BOX B_SINGLE .
                  PICTURE CAPS ;
                VALID ( Empty( mLocation ) .or. ;
                     ( Help_Fill( @mLocation, "Location", flds_2 ;
                           ,, LocFilter,,,,, Loc_Header, ;
                           { || GetOrder("LOCATION") ) ;
                           ,, 1, 1 ) .and. ;
          Pro_Loc( mLocation, @mDistrict, 6, 56, cProvince, 07, 56 ) ) );
                  HELP "Any key + <ENTER> for picklist"

@ 6,56 Get mDISTRICT BOX B_SINGLE .
                  PICTURE CAPS .
                WHEN Empty( mLocation ) .
                VALID Help_Fill( @mDistrict, "Location", flds_5 ;
                           ,,,,,,, Dis_Header, ;
                           { || GetOrder("DISTRICT") ) ;
                           ,, 1, 2 ) .and. ;
            Pro_Dist( mDistrict, 07, 56 ) // paint province at 07,56

@ 9,56 Get mMAP_SHEET PICTURE CAPS
@10,56 Get mGRID_REF PICTURE CAPS

@11,58 Say Space(11)
@11,58 Get mLATITUDE  PICTURE  "@R 99° 99.99'" ;
                VALID mLatitude >= 0 .and. mLatitude <= 9000
@11,Col() + 1 Get cNS_Lat PICTURE CAPS ;
                   VALID cNS_Lat $ "NS"

@12,57 Say Space(11)
@12,57 Get mLONGITUDE PICTURE "@R 999° 99.99'" ;
               VALID mLongitude >= 0 .and. mLongitude <= 18000
@12,Col() + 1 Get cEW_Long PICTURE CAPS ;
                   VALID cEW_Long $ "EW"


@15,55 Get mTYPES PICTURE "@S22!"
@17,55 Get mNO_MINES PICTURE [999] ;
                VALID mNO_MINES > 0 ;
                HELP "Approximate number of mines (1-999)"

@17,59 Get mNUMBERS PICTURE "@S18!" ;
                HELP "Qualifier on number of mines"

@19,55 Get mLAID_BY PICTURE "@S22!"
@21,55 Get mDATE_LAID PICTURE "@S22!"

lResult := Read_It( ,,getList )

If lResult
   mLatitude  := LatDec( mLatitude, cNS_Lat )
   mLongitude := LongDec( mLongitude, cEW_Long )
   mTag   := T
```

```
      If M->Is_Append
        @ 5,15 Say ( nReport_No := Next( "Report_No" )) PICTURE "@9"
      EndIf
    EndIf

Return lResult


* ============================================================
FUNCTION MAS_SAY
* ============================================================
Local lResult := .T.
Local lMemo   := .F.
Local Ass_Rec, Mine_Rec
Local nDets
Local GetList := {}

   Ass_Rec := RECNO()
   SELECT Mines
   Mine_Rec := RECNO()                      // save record number
   COUNT TO nDets WHILE Mines->Report_No = Mine_Ass->Report_No
   GO Mine_Rec
   SELECT Mine_Ass
   GO Ass_Rec

   SysColor( 2 )                                    // hilighted video
   * Set Keys for tagging and untagging
   SET KEY K_ALT_A TO TAG_RECORD            // Menu to Tag or Untag all
   SET KEY K_ALT_T TO TAG_RECORD            // Tag a record
   SET KEY K_ALT_U TO TAG_RECORD                  // Untag a record
   SET KEY K_ALT_X TO Imm_Exit              // Exit from program

   SYSCOLOR ( "N/W" )
   @ 5,15 Say Transform( Val( Mine_Ass->Report_No), "@Z" )
   @ 6,15 Say MINE_ASS->REP_DATE
   @ 7,15 Say Proper( MINE_ASS->REP_NAME )

   @ 5,56 Say Proper( MINE_ASS->LOCATION )
   @ 6,56 Say Proper( MINE_ASS->DISTRICT )
   @ 7,56 Say Proper( Fetch( Mine_Ass->District, "Location", "Province", 2 ) )

   @ 9,56 Say MINE_ASS->MAP_SHEET
   @10,56 Say MINE_ASS->GRID_REF

   DispLatLong( Mine_Ass->LATITUDE, 11, 57, Mine_Ass->Longitude, 12, 56 )

   @12,06 Say nDets PICTURE [99]
   @15,55 Say SubStr( MINE_ASS->TYPES, 1, 22 )
   @17,55 Say MINE_ASS->NO_MINES
   @17,59 Say Proper( SubStr( MINE_ASS->NUMBERS, 1, 18 ) )
   @19,55 Say Proper( SubStr( MINE_ASS->LAID_BY, 1, 22 ) )
   @21,55 Say Proper( SubStr( MINE_ASS->DATE_LAID, 1, 22 ) )

   SysColor( "N/W" )
   KEYBOARD CHR( K_ESC )
   MemoEdit(Mine_Ass->Source, 10, 14, 12, 40, .F.)

   KEYBOARD CHR( K_ESC )
   MemoEdit(Mine_Ass->Reported, 15, 04, 16, 40, .F.)

   KEYBOARD CHR( K_ESC )
   MemoEdit(Mine_Ass->Suspected, 18, 04, 19, 40, .F.)

   KEYBOARD CHR( K_ESC )
   MemoEdit(Mine_Ass->Remarks, 21, 04, 21, 40, .F.)
```

```
Return lResult


* ============================================================
FUNCTION MIN_SCR
* ============================================================
Local lResult := T

//   SysColor( AACLR_BOX )
   BOXER( 9, 3, 22, 76, 1,,, "GR+/BR" ,, .T. )
   SysColor( "GR+/BR" )
   @ 9,21 Say "┬"
   @ 9,57 Say "┬"
   @ 9,64 Say "┬"
   @14, 3 Say "├"
   @14,76 Say "┤"
   @10,21 Say "│                                                    │"
   @11,21 Say "│   Mines Found    │ Mines Cleared │                 │"
   @12,21 Say "├──────────────────┼───────────────┤                 │ Incident"
   @13, 5 Say "Mine Type          | No. |   Date   | No. |   Date   | Memo |  Number"
   @14, 4 Say
"└──────────────────┴─────┴──────────┴─────┴──────────┴──────┴──────────┘"

   SysColor( AACLR_HEADER )
   @ 9,30 CLEAR TO  9,50
   @ 9,31 Say "L A N D   M I N E S"

Return lResult


* ============================================================
FUNCTION MINE_ADD
* ============================================================
RETURN MenuAddEdit( 1 )


* ============================================================
FUNCTION MINE_DUP
* ============================================================
RETURN MenuAddEdit( 3 )          // temporarily same as an add - see below


* ============================================================
FUNCTION MINE_EDIT
* ============================================================
RETURN MenuAddEdit( 0 )


* ============================================================
FUNCTION MenuAddEdit( nMode )   // Menus to add/dup/edit Mines
* ============================================================
MemVar cols, graf_row, choices
LOCAL cPrefix   := IF( nMode == 1, "Add a ", ;
                   IIF( nMode == 0, "Edit a ", "Add a " ) )     // Add is dup
LOCAL cColFind  := IF( nMode == 1, "Add ", ;
                   IIF( nMode == 0, "Edit ", "Dup " ) )

LOCAL aMineMenu := { cPrefix + "mine detail" }
LOCAL udf_ra    := { { || MineScroll( nMode ) } }

   nMode := IIF( nMode == 3, 1, nMode ) // workaround for dup

   PULL_OPEN( aMineMenu, graf_row + 1, ;
              cols[ fSCAN( choices, " " + cColFind ) ], 1, .T., udf_ra )

RETURN .T.
```

```
* ═══════════════════════════════════════════════════════════════
FUNCTION MineScroll( nMode )
* ═══════════════════════════════════════════════════════════════
Mines->( DB_SCROLL( 15, 4, 21, 74, ;
                    { | n, | Mine_RSay( n, | ) }, ;
                    MINE_ASS->Report_No, ;
                    { | m, n | Mine_RGet( m, n ) }, ;
                    .F., .F., nMode + 2 ) )
StatusLine()

RETURN .T.


* ═══════════════════════════════════════════════════════════════
FUNCTION MIN_KEY( aRecord )      // dummy function
* ═══════════════════════════════════════════════════════════════
Return .T.


* ═══════════════════════════════════════════════════════════════
FUNCTION MIN_FLD          // dummy function
* ═══════════════════════════════════════════════════════════════
Return .T.


* ═══════════════════════════════════════════════════════════════
FUNCTION MIN_SAY
* ═══════════════════════════════════════════════════════════════

   * Set Keys for tagging and untagging
   SET KEY K_ALT_A TO TAG_RECORD            // Menu to Tag or Untag all
   SET KEY K_ALT_T TO TAG_RECORD            // Tag a record
   SET KEY K_ALT_U TO TAG_RECORD                  // Untag a record
   SET KEY K_ALT_X TO Imm_Exit              // Exit from program

   // special scroll box for mine types and their data

   Mines->(DB_Scroll(15, 04, 21, 75, ;
                           { | n, | Mine_RSay( n, | ) }, ;
                           Mine_Ass->Report_No,,,, 1 ))
Return .T.


* ═══════════════════════════════════════════════════════════════
FUNCTION Mine_RSay( nRow, lBool)
* ═══════════════════════════════════════════════════════════════
LOCAL cOldClr := If( lBool, ;
                  SetColor( PartColor( 2, GetColor( 2 ), .F. )), ;
                  SetColor() ;
                )
Local lMemo       := MemoStatus(Mines->Remarks)

   @ nRow, 05 Say Proper( Mines->Mine_Type )
   @ nRow, 23 Say Mines->Quantity
   @ nRow, 30 Say Mines->Date_Laid
   @ nRow, 41 Say Mines->Qnty_Clear
   @ nRow, 48 Say Mines->Date_Clear
   @ nRow, 61 Say LTOC( lMemo )
   @ nRow, 68 Say Mines->Inc_Number
   SetColor( cOldClr )

Return .T.
```

```
* ═══════════════════════════════════════════════════════════════
FUNCTION Mine_RGet( nRow, nMode)
* ═══════════════════════════════════════════════════════════════
Return UGet( nMode, { | a | Mine_Gets( nRow, a ) } )


* ═══════════════════════════════════════════════════════════════
FUNCTION Mine_Gets( nRow, aRecord )
* ═══════════════════════════════════════════════════════════════
Local lResult
Local GetList := {}
Local Flds_7     := { "MINE_TYPE", "DESCRIPT" }, ; // Mine Type array
      Min_Header := { "Mine Type", "Description" }

Local mMine_Type := M( Mine_Type )

   If M->Is_Append
      M( Unique_No )  := "new "
      M( Entry_Date ) := Date()
      M( Report_No )  := Mine_Ass->Report_No
   Else
//       Aud_Before()
   EndIF

   @ nRow, 05 Get mMine_Type BOX B_SINGLE ;
                                PICTURE CAPS ;
                            VALID Help_Fill( @mMine_Type, ;
                                        "Mine_Typ", ;
                                        Flds_7 ;
                                        ,,,,,,, Min_Header,,, 1, 1 )

   @ nRow, 23 Get M( Quantity ) PICTURE [9999]
   @ nRow, 30 Get M( Date_Laid ) PICTURE "@D"
   @ nRow, 41 Get M( Qnty_Clear ) PICTURE [9999]
   @ nRow, 48 Get M( Date_Clear ) PICTURE "@D" ;
                             WHEN M( Qnty_Clear ) > 0
   @ nRow, 58 Get M( Remarks ) MEMO
   @ nRow, 68 Get M( Inc_Number )

   IF ( lResult := READ_IT( .T., , getList ) )
      M( Tag ) := .T.                  // TAG true If edit or add
      M( Mine_Type ) := mMine_Type
      If M->Is_Append
        M( Unique_No ) := Next( "UNIQUE_NO" )
      Else
        If UpDated()
//          Aud_After()          // Check Audit Trail for changes
        EndIf
      EndIf
   ENDIF

Return lResult

* ═══════════════════════════════════════════════════════════════ *
EXTERNAL SET_TABLE, ;                 // THE_USUAL()'s default Init udf
       RPT_MENU, ;                   // Reports interface
       UQR_SCR, ;                    // Queries table maintenance
       SYS_SCR, ;                    // System configuration
       AUTO_INIT, ;                  // AUTO module
       REPLACE_EM, RPLCHIFLDS, ;     // Force in multi-user versions
       GET_STD2, UGET, DUST_1        // Ditto
External DispLatLong, LatDeg, LongDeg, LatDec, LongDec
* ═══════════════════════════════════════════════════════════════ *
```

```
*
*  Program          SHA_MIG.PRG
*  System           ARTFUL.LIB for Clipper 5.0
*  Functions        Mig_Scr, Mig_Key, Mig_Fld, Mig_Say, Pro_Dist
*                   Ppl_Scr, Ppl_Key, Ppl_Fld, Ppl_Say
*                   Res_Scr, Res_Key, Res_Fld, Res_Say
*  Purpose          Maintain an audit trail of all changes
*  Conventions      Note:
*  Syntax
*  Parameters       None
*  Example
*  Assumes
*  Side Effects
*  Returns
*
*  ARTFUL Calls     THE_USUAL, BOXER, SYSCOLOR, UGET, READ_IT
*  Authors          David W. Zimmerly
*  Copyright        BlueWater Software 1991- 1993
*

#include "artful.ch"
#include "box.ch"
#include "set.ch"
#include "aa_pulld.ch"
#include "aa_gets.ch"

#xtranslate M( <foo> )    => aRecord\[ FieldPos( <"foo"> ) \]
#xtranslate MStr( <foo> ) => aRecord\[ FieldPos( <foo> ) \]

#define caps      '@K!'
#define cont_pic  '@K! 9999999XXXXXXXXXXX'
#define AACLR_BOX "GR+/B"

/*
* ============================================================================
FUNCTION MIG_SCR
* ============================================================================
   SysColor( AACLR_BOX )
   BOXER( 3, 2, 23, 77, 2,,, "GR+/B" ,, .T. )
   @ 8, 2 Say "├"
   @ 8,77 Say "┤"
   @19, 2 Say "├"
   @19,77 Say "┤"
   @23,19 Say "┴"
   @ 8, 3 Say
"──────────────────────────────────────────────────────────────────────────"
   @19, 3 Say
"──────────────────┘                                                       "

   SysColor( AACLR_BOX )

   @ 4,51 Say "1980 Population:"
   @ 5, 5 Say "District:                                    Oct 1992 Population:"
   @ 6, 5 Say "Province:                                     Current Population:"
   @ 6,48 Say "Current Population:"
   @ 7, 5 Say "Migration Status:          % completed    Current Beneficiaries:"
   @ 9, 5 Say "Memo:"
   @11,11 Say "DISTRICT MIGRATION"
   @12,10 Say "──────────────────────"
   @13,10 Say "     Within district:"
   @14,15 Say "From in-country:"
   @15,11 Say "From out of country:"
   @16,17 Say "Out migration:"
   @20,19 Say "|                          Current Population:"
   @21, 5 Say "Calculations  |   Population May 1995 (- natural increase):"
   @22,19 Say "|   Population May 1995 (+ natural increase):"
```

```
   SysColor( AACLR_HEADER )
   @ 3,31 CLEAR TO  3,49
   @ 3,32 Say "M I G R A T I O N"

   SysColor( AACLR_BOX )
   BOXER( 9, 32, 17, 70, 4,, "GR+/BG" ,, .T. )
//   SysColor( AACLR_BOX )
   SysColor( "GR+/BG" )
   @ 9,45 Say "╥"
   @ 9,58 Say "╥"
   @12,32 Say "╟"
   @12,70 Say "╢"
   @17,45 Say "╨"
   @17,58 Say "╨"

   @10,34 Say "Oct '92 to ▌ Present to ▌ Migration"
   @11,36 Say "Present  ▌   May '95  ▌ Completed"
   @12,33 Say "═══════ ╪ ═══════ ╪ ═══════"
   @13,45 Say "▌               ▌       ▒"
   @14,45 Say "▌               ▌       ▒"
   @15,45 Say "▌               ▌       ▒"
   @16,45 Say "▌               ▌       ▒"

   SysColor( AACLR_NORMAL )
RETURN .T.
```

```
* ═══════════════════════════════════════════════════════════
FUNCTION MIG_KEY
* ═══════════════════════════════════════════════════════════
Local lResult    := .F.
Local flds_5     := { "DISTRICT", "PROVINCE" }, ; // District array
      Dis_Header := { "District", "Province" } // District header

Local GetList    := {}


   SysColor(2)     // Gets

   If mPermission <> "EXPERT"
      RETURN .F.
   EndIf

   mTag := .T.              // TAG true if edit or add

   If M->Is_Append
   Else
   EndIF

   @ 5,15 Get mDISTRICT BOX B_SINGLE ;
                        PICTURE caps ;
                     VALID Help_Fill( @mDistrict, "Location", flds_5 ;
                                    ,,,,,,, Dis_Header, ;
                                    { || GetOrder("DISTRICT") } ;
                                    ,, 1, 2 ) .and. ;
                  Pro_Dist( mDistrict, 6, 15 )

RETURN Read_It( ,,Getlist)


* ═══════════════════════════════════════════════════════════
FUNCTION MIG_FLD
* ═══════════════════════════════════════════════════════════
```

```
Local lResult    := .F.
Local GetList    := {}

   SysColor(2)     // Gets

   If M->Is_Append
   Else
      Aud_Before()
   Endif

   If mPermission <> "EXPERT"
      RETURN .F.
   EndIf

   @13,36 Get mA_RESETTLE PICTURE [999999]
   @13,49 Get mB_RESETTLE PICTURE [999999]
   @14,36 Get mARRA_DIST PICTURE [999999]
   @14,49 Get mARRB_DIST PICTURE [999999]
   @15,36 Get mARRA_COUN PICTURE [999999]
   @15,49 Get mARRB_COUN PICTURE [999999]
   @16,36 Get mOUTA_MIGR PICTURE [999999]
   @16,49 Get mOUTB_MIGR PICTURE [999999]

   lResult := Read_It( ,,GetList )

   If lResult
      If M->Is_Append
      Else
        If UpDated()
           Aud_After()          // Check Audit Trail for changes
        EndIf
      EndIf
   EndIf

RETURN lResult


* ============================================================
FUNCTION MIG_Say
* ============================================================
Local lMemo      := .F.
Local GetList    := {}

Local CalcDisMig  := 0, ;
      Per_Res     := 0, ;
      PerDis      := 0, ;
      PerRef      := 0, ;
      PerOut      := 0, ;
      CalcPop     := 0, ;
      Calc95Minus := 0, ;
      Calc95Plus  := 0

   SysColor( 2 )                                  // hilighted video
   * Set Keys for tagging and untagging
   SET KEY K_ALT_A TO TAG_RECORD          // Menu to Tag or Untag all
   SET KEY K_ALT_T TO TAG_RECORD          // Tag a record
   SET KEY K_ALT_U TO TAG_RECORD                 // Untag a record
   SET KEY K_ALT_X TO Imm_Exit            // Exit from program

   SYSCOLOR ( "W+/W" )
   @ 5,15 Say MIGRATE->DISTRICT
   @ 6,15 Say Proper( Fetch( MIGRATE->District, "Location", ;
                        "Province", "DISTRICT" ) )
   @ 7,23 Say CalcDisMig PICTURE [999]
```

```
   @ 4,68 Say POPULATE->POP_1980
   @ 5,68 Say POPULATE->POP_OCT92
   @ 6,68 Say POPULATE->CURR_POP
   @ 7,68 Say POPULATE->CURR_BENEF

   @13,36 Say MIGRATE->A_RESETTLE PICTURE [999999]
   @13,49 Say MIGRATE->B_RESETTLE PICTURE [999999]
   @14,36 Say MIGRATE->ARRA_DIST PICTURE [999999]
   @14,49 Say MIGRATE->ARRB_DIST PICTURE [999999]
   @15,36 Say MIGRATE->ARRA_COUN PICTURE [999999]
   @15,49 Say MIGRATE->ARRB_COUN PICTURE [999999]
   @16,36 Say MIGRATE->OUTA_MIGR PICTURE [999999]
   @16,49 Say MIGRATE->OUTB_MIGR PICTURE [999999]
   @13,62 Say Per_Res PICTURE [999]
   @14,62 Say PerDis PICTURE [999]
   @15,62 Say PerRef PICTURE [999]
   @16,62 Say PerOut PICTURE [999]
   @20,65 Say CalcPop PICTURE [999999]
   @21,65 Say Calc95Minus PICTURE [999999]
   @22,65 Say Calc95Plus PICTURE [999999]

   * Determine If memo field has an entry
   lMemo := MemoStatus(Migrate->Remarks)
   @ 09, 11 CheckBox lMemo

RETURN .T.

*/

* ============================================================
FUNCTION PRO_DIST( cDistrict, nRow, nCol )
* ============================================================
Local cAlias := Select()
Local nOldOrder := IndexOrd()
Local cProvince
   If Seek_Rel( cDistrict, "Location", 2 )
      cProvince := Location->Province
      Select Location
      SetOrder( "District" )
      Lookup( cDistrict, "Location", "Proper(Province)", nRow, nCol )
   Else
      Pop_Msg( "cDistrict not found" )
   EndIf
   Select( cAlias )
   Set Order To nOldOrder
Return .T.



* ============================================================
FUNCTION PPL_SCR
* ============================================================
   SysColor( AACLR_BOX )
   BOXER( 5, 15, 21, 64, 2,,, "GR+/B" ,, .T. )

   @10,15 Say "╟"
   @10,64 Say "╢"
   @12,15 Say "╟"
   @12,64 Say "╢"
   @18,15 Say "╟"
   @18,64 Say "╢"
   @10,16 Say "──────────────────────────────────────────────"
   @12,16 Say "──────────────────────────────────────────────"
   @18,16 Say "──────────────────────────────────────────────"

   SysColor( AACLR_NORMAL )
```

```
   @ 7,18 Say Replicate( " ", 10 - Len(Trim(Defaults->District ))) + ,
              Proper(Trim(Defaults->District)) + ":"
   @ 7,53 Say "Memo:"
   @ 8,18 Say Replicate( " ", 10 - Len(Trim(Defaults->Province))) + ,
              Proper(Trim(Defaults->Province))+":"
   @ 9,20 Say "Current Beneficiaries:"
   @11,23 Say "Current Population:"
   @13,20 Say Replicate( " ", 22 - Len(Trim(Defaults->Age_1_Hdr))) + ,
              Trim(Defaults->Age_1_Hdr)
   @14,20 Say Replicate( " ", 22 - Len(Trim(Defaults->Age_2_Hdr))) + ,
              Trim(Defaults->Age_2_Hdr)
   @15,20 Say Replicate( " ", 22 - Len(Trim(Defaults->Age_3_Hdr))) + ,
              Trim(Defaults->Age_3_Hdr)
   @16,20 Say Replicate( " ", 22 - Len(Trim(Defaults->Age_4_Hdr))) + ,
              Trim(Defaults->Age_4_Hdr)
   @17,20 Say Replicate( " ", 22 - Len(Trim(Defaults->Age_5_Hdr))) + ,
              Trim(Defaults->Age_5_Hdr)

   @19,19 Say Replicate( " ", 22 - Len(Trim(Defaults->Pop_1_Hdr))) + ,
              Trim(Defaults->Pop_1_Hdr) + ":"
   @20,19 Say Replicate( " ", 22 - Len(Trim(Defaults->Pop_2_Hdr))) + ,
              Trim(Defaults->Pop_2_Hdr) + ":"

   SysColor( AACLR_HEADER )
   @ 5,21 CLEAR TO  5,59
   @ 5,22 Say Expand( Trim( Defaults->District )) + "   P O P U L A T I O N"

   SysColor( AACLR_NORMAL )

RETURN .T.


* ============================================================
FUNCTION PPL_KEY
* ============================================================
MemVar mTag, mPermission, mDistrict
Local lResult    := .F.
Local flds_5     := { "DISTRICT", "PROVINCE" }, ; // District array
      Dis_Header := { Proper(Trim(Defaults->District)), ;
                      Proper(Trim(Defaults->Province)) } // District header

Local GetList    := {}


   SysColor(2)     // Gets

   If mPermission <> "EXPERT"
      RETURN .F.
   EndIf

   mTag := .T.             // TAG true If edit or add

   @ 7,30 Get mDISTRICT BOX B_SINGLE ;
                        PICTURE caps ;
                     VALID Help_Fill( @mDistrict, "Location", flds_5 ;
                                 ,,,,,,, Dis_Header, ;
                                 ( !! GetOrder("DISTRICT") ) ;
                                 ,, 1, 2 ) .and. ;
                               Unique( mDistrict ) .and. ;
                               Pro_Dist( mDistrict, 8, 30 )

RETURN Read_It( ,,Getlist)


* ============================================================
FUNCTION PPL_FLD
```

```
*
MemVar mPermission
Local lResult       F
Local GetList       {}

   SysColor(2)     // Gets

   If M->Is_Append
   Else
      Aud_Before()
   EndIf

   If mPermission <> "EXPERT"
      RETURN  F
   EndIf

   @ 9,43 Get mCURR_BENEF PICTURE [9999999]
   @11,43 Get mCURR_POP PICTURE [9999999]

   @13,43 Get mAGENCY1POP PICTURE [9999999]
   @14,43 Get mAGENCY2POP PICTURE [9999999]
   @15,43 Get mAGENCY3POP PICTURE [9999999]
   @16,43 Get mAGENCY4POP PICTURE [9999999]
   @17,43 Get mAGENCY5POP PICTURE [9999999]

   @19,43 Get mPOP_1 PICTURE [9999999]
   @20,43 Get mPOP_2 PICTURE [9999999]

   lResult := Read_It( ,,GetList )

   If lResult
      If M->Is_Append
      Else
        If UpDated()
           Aud_After()          // Check Audit Trail for changes
        EndIf
      EndIf
   EndIf

RETURN lResult


*
FUNCTION PPL_Say
*

Local lMemo     := .F.
Local GetList := {}

   SysColor( 2 )                                  // hilighted video
   * Set Keys for tagging and untagging
   SET KEY K_ALT_A TO TAG_RECORD          // Menu to Tag or Untag all
   SET KEY K_ALT_T TO TAG_RECORD          // Tag a record
   SET KEY K_ALT_U TO TAG_RECORD                // Untag a record
   SET KEY K_ALT_X TO Imm_Exit            // Exit from program

   SYSCOLOR ( "W+/W" )

   @ 7,30 Say POPULATE->DISTRICT
   @ 8,30 Say Proper( Fetch( POPULATE->District, "Location", ;
                             "Province", "DISTRICT" ) )
   @ 9,43 Say POPULATE->CURR_BENEF PICTURE [9999999]

   @11,43 Say POPULATE->CURR_POP  PICTURE [9999999]

   @13,43 Say POPULATE->AGENCY1POP PICTURE [9999999]
```

```
   @14,43 Say POPULATE->AGENCY2POP PICTURE [9999999]
   @15,43 Say POPULATE->AGENCY3POP PICTURE [9999999]
   @16,43 Say POPULATE->AGENCY4POP PICTURE [9999999]
   @17,43 Say POPULATE->AGENCY5POP PICTURE [9999999]
   @19,43 Say POPULATE->POP_1   PICTURE [9999999]
   @20,43 Say POPULATE->POP_2  PICTURE [9999999]

   * Determine If memo field has an entry
   lMemo := MemoStatus(Populate->Remarks)
   @ 07, 59 CheckBox lMemo

RETURN .T.


* ============================================================
FUNCTION RES_SCR
* ============================================================
   SysColor( AACLR_BOX )
   BOXER( 3, 6, 22, 74, 2,,, "GR+/BR" ,,  T  )

   SysColor( "GR+/BR" )
   @ 7, 6 Say "╟"
   @ 7,74 Say "╢"
   @11, 6 Say "╟"
   @11,74 Say "╢"
   @18, 6 Say "╟"
   @18,74 Say "╢"

   @ 7, 7 Say "─ ──────────────────────────────────────────────────────────────────"
   @11, 7 Say "──────────────────────────────────────────────────────────────────"
   @18, 7 Say "─        ──────────────────────────────────────────────────────────"

   @ 5, 9 Say "Resource No                                    Report Date "
   @ 8,14 Say "Sector:                                Qualifier:"
   @ 9,16 Say "Type:                                   Number:"
   @12,12 Say "Location                                        ddd° mm.mm' S"
   @13,10 Say Replicate( " ", 10 - Len(Trim(Defaults->District))) + ;
              Proper(Trim(Defaults->District))+":"
   @13,49 Say "Latitude:"
   @14,10 Say Replicate( " ", 10 - Len(Trim(Defaults->Province))) + ;
              Proper(Trim(Defaults->Province))+":"
   @14,48 Say "Longitude:"
   @15, 9 Say "Location"
   @16, 9 Say "Description:"

   SysColor( AACLR_HEADER )
   @ 3,22 CLEAR TO  3,58
   @ 3,23 Say "S E C T O R A L   R E S O U R C E S"
   @ 7, 9 CLEAR TO  7,16
   @ 7, 9 Say "Resource"
   @11, 9 CLEAR TO 11,16
   @11, 9 Say "Location"
   @18, 9 CLEAR TO 18,15
   @18, 9 Say "Remarks"

   SysColor( AACLR_NORMAL )
RETURN .T.


* ============================================================
FUNCTION RES_KEY
* ============================================================
MemVar mPermission, mRes_Num
Local lResult := .T.
Local GetList := {}
```

```
   SysColor(2)     // Gets

   If mPermission <> "EXPERT"
      RETURN  F
   EndIf

   If M->Is_Append
      mRes_Num  := " new "
   Else
   EndIF

   @ 5,22 Say mRes_Num
   SysColor(1)          // Normal

RETURN lResult


* ============================================================
FUNCTION RES_FLD
* ============================================================
MemVar mRep_Date, mRemarks, mDistrict, mLatitude, mLongitude
MemVar mPermission, mSector, mType, mLocation, mRes_Num, mNo_Func

Local lResult        := F
Local lMemo          := F
Local cDistrict      := Space( 20 )
Local cProvince      := Space( 20 )

Local aSectorList    := { "HEALTH", ;
                          "WATER", ;
                          "EDUCATION" ;
                        }
Local aTypeList      := { "SCHOOL", ;
                          "HEALTH POST/CENTER", ;
                          "HOSPITAL", ;
                          "WELLS & BOREHOLES" ;
                        }

Local flds_2         := { "LOCATION", "Proper( DISTRICT )", "Proper( PROVINCE )", ;
                          "Proper( TYPE )" }, ; // Location array
      Loc_Header     := { "Location", Proper(Trim(Defaults->District)), ;
                          Proper(Trim(Defaults->Province)), "Type" }    // Location header

Local flds_5         := { "DISTRICT", "PROVINCE" }, ;       // District array
      Dis_Header     := { Proper(Trim(Defaults->District)), ;
                          Proper(Trim(Defaults->Province)) }       // District header
Local flds_10        := { "SECTOR", "TYPE" }, ;             // Resource array
      Res_Header     := { "Resource:Sector", "Resource:Type" } // Resource header
Local flds_11        := { "SECTOR" }, ;                     // Sector array
      Sec_Header     := { "Resource:Sector" }               // Sector header

Local LocFilter      := "Location->Location != ' ' "
Local ResFilter      := "Res_Type->Sector == mSector"
Local cNS_Lat        := "S", ;
      cEW_Long       := "E"

Local GetList        := {}

   SysColor(2)     // Gets

   If mPermission <> "EXPERT"
      RETURN .F.
   EndIf

If M->Is_Append
      mRep_Date     := Date()
```

```
      mRemarks    := ""
      mDistrict   := Space( 20 )
      mLatitude   := 0.00
      mLongitude  := 0.00
      cNS_Lat     := "S"
      cEW_Long    := "E"
   Else
      mLatitude   := LatDeg( Resource->Latitude, @cNS_Lat )
      mLongitude  := LongDeg( Resource->Longitude, @cEW_Long )
   EndIf

   @ 5,64 Get mREP_DATE PICTURE "@D"

   @ 8,22 Get mSECTOR BOX B_SINGLE ;
                     PICTURE caps ;
                  VALID Help_Fill( @mSector, "Res_Type", flds_11 ;
                                 ,,,,,,, Sec_Header, ;
                                 { || GetOrder("SECTOR") } ;
                                 ,, 1, 2 )

// ResFilter := "Res_Type->Sector = mSector"
   @ 9,22 Get mTYPE BOX B_SINGLE ;
                  PICTURE caps ;
               VALID Help_Fill( @mType, "Res_Type", flds_10 ;
                                 ,, ResFilter,,,,, Res_Header, ;
                                 { || GetOrder("SECTOR+TYPE") } ;
                                 ,, 2, 1 )
   @ 8,57 Get mQUALIFIER
   @ 9,57 Get mNO_FUNC PICTURE [99999] ;
                     Valid mNo_Func > 0

   @12,22 Get mLOCATION ;    // BOX B_SINGLE ;
                     PICTURE caps ;
                  VALID Empty( mLocation ) .or. ;
                        ( Help_Fill( @mLocation, "Location", flds_2 ;
                                 ,, LocFilter,,,,, Loc_Header, ;
                                 { || GetOrder("LOCATION") } ;
                                 ,, 1, 1 ) .and. ;
          Pro_Loc( mLocation, @mDistrict, 13, 22, cProvince, 14, 22 ) ) ;
                     HELP "Any key + <ENTER> for picklist"


   @13,22 Get mDISTRICT BOX B_SINGLE ;
                     PICTURE caps ;
                  WHEN Empty( mLocation ) ;
                  VALID Help_Fill( @mDistrict, "Location", flds_5 ;
                                 ,,,,,,, Dis_Header, ;
                                 { || GetOrder("DISTRICT") } ;
                                 ,, 1, 2 ) .and. ;
               Pro_Dist( mDistrict, 14, 22 ) // paint province at 14,22

   @13,59 Get mLATITUDE PICTURE "@R  99° 99.99'" ;
                  VALID mLatitude >= 0 .and. mLatitude <= 9000

   @13,Col() + 1 Get cNS_Lat PICTURE caps ;
                        VALID cNS_Lat $ "NS"

   @14,58 Get mLONGITUDE PICTURE "@R  999° 99.99'" ;
                  VALID mLongitude >= 0 .and. mLongitude <= 18000

   @14,Col() + 1 Get cEW_Long PICTURE caps ;
                        VALID cEW_Long $ "EW"

   @16,22 Get mLOC_DESC

   lResult := Read_It( ,,GetList )
```

```
    If lResult
       mLatitude     := LatDec( mLatitude, cNS_Lat )
       mLongitude    := LongDec( mLongitude, cEW_Long )

       If M->Is_Append
          @ 5,22 Say ( mRes_Num    := Next( "Res_Num" )) PICTURE "@9"
       EndIf
    EndIf

RETURN lResult


*  ---------------------------------------------------------------------
FUNCTION RES_Say
*  ---------------------------------------------------------------------
Local lMemo     := F
Local GetList   := {}

   SysColor( 2 )                                    // hilighted video
   * Set Keys for tagging and untagging
   SET KEY K_ALT_A TO TAG_RECORD            // Menu to Tag or Untag all
   SET KEY K_ALT_T TO TAG_RECORD            // Tag a record
   SET KEY K_ALT_U TO TAG_RECORD                   // Untag a record
   SET KEY K_ALT_X TO Imm_Exit              // Exit from program

   SYSCOLOR ( "N/W" )

   @ 5,22 Say Transform( Val( Resource->Res_Num), "@Z" )

   @ 5,64 Say RESOURCE->REP_DATE
   @ 8,22 Say Proper( RESOURCE->SECTOR )
   @ 8,57 Say Proper( RESOURCE->QUALIFIER )
   @ 9,22 Say Proper( RESOURCE->TYPE )
   @ 9,57 Say RESOURCE->NO_FUNC PICTURE [99999]

   @12,22 Say Proper( RESOURCE->LOCATION )
   @13,22 Say Proper( RESOURCE->DISTRICT )
   @14,22 Say Proper( Fetch( Resource->District, "Location", "Province", 2 ) )

   DispLatLong( Resource->LATITUDE, 13, 59, Resource->Longitude, 14, 58 )

   @16,22 Say RESOURCE->LOC_DESC
   SysColor( "N/W" )
   KEYBOARD CHR( K_ESC )
   MemoEdit(Resource->Remarks, 19, 09, 21, 71, .F.)

RETURN .T.


*  ---------------------------------------------------------------------  *

EXTERNAL SET_TABLE, ;                    // THE_USUAL()'s default Init udf
         RPT_MENU, ;                    // Reports interface
         UQR_SCR, ;                     // Queries table maintenance
         SYS_SCR, ;                     // System configuration
         AUTO_INIT, ;                   // AUTO module
         REPLACE_EM, RPLCHIFLDS, ;      // Force in multi-user versions
         Get_STD2, UGet, DUST_1         // Ditto

External DispLatLong, LatDeg, LongDeg, LatDec, LongDec

*  ---------------------------------------------------------------------  *
```

```
/*
   Program:         C:\CLIPPERS\ARTFUL\SHAMAN\ACTTITLE.PRG
   Functions:       Att_Scr()
                    Att_Say()
                    Att_Key()
                    Att_Fld()
   Purpose:         Form for The_Usual()
   Table:           ACTTITLE
   Written by:      Artful WinGen
   Date:            Dec 12, 1993
   Template:        WINUSUAL.ATM
   Compile with:    /l/n/w
   Link with:       Artful Two v2.75 or later.
   Test:            Make with: MM <progName> /b/n/w/dDEBUG

   Functions:
     Att_Test():      Tester
     Att_Scr():       Paint screen for ACTTITLE
     Att_Say():       Say current record of ACTTITLE
     Att_Key():       Execute Gets for ACTTITLE
     Att_Fld():       Required for The_Usual()

   NOTE:
     If your form references other tables,
     add Db_Open() calls for them to Att_Test()

   To incorporate this file in your application...
     - Set Id_Char to <Att> in the DICT record
       for the ACTTITLE table.
     - Set DICT->Udf_Init to <SET_TABLE>, or to the name
       of your custom menu udf for The_Usual().
     - Add the name of this file to your application .CLP file.

 */

#include "artful.ch"
#include "aa_gets.ch"
// #include "aa_mouse.ch"


/*
  ┌──────────────────────────────────────────────────┐
  │ The following #defines must be set BEFORE WinGen │
  │ inserts code for your window into this template: │
  └──────────────────────────────────────────────────┘
 */

// Include field declarations.
#define AA_FIELDEFS

// Comment out to obtain your own window colours:
// #define AA_SYSCOLOR

// Suppress field-length "XXX..." SAY pictures:
#define AA_NOSAYXXX

// Suppress field-length "XXX    " GET pictures:
#define AA_NOGETXXX


/*
  ┌──────────────────────────────────────────────────┐
  │ The following #defines take effect AFTER code generation: │
  └──────────────────────────────────────────────────┘
 */
```

```
// Box-wide headers, footers?
#define WIN_STYLE  T


// If this is one of multiple screens in a form, uncomment this line:
// #define AA_MULTISCR

// NOTE  Uncomment this if you have AAMOUSE.LIB:
// #define AA_MOUSE


// Tokens required by WinGen:
#define CALIAS          "NEEDS"
#define FORMTABLES      { { 'NEEDS.DBF', 'NEEDS' }, { 'ACTTITLE.DBF', 'ACTTITLE' } }
#define NTOP            6
#define NLEFT           4
#define NBOTTOM         23
#define NRIGHT          75
#define CWINCOLOR       'B/W'
#define XFRAME          '┌─┐│┘─└│'
#define CFILLCHAR       ' '
#define CFRAMECLR       'BR/W*'
#define CTITLE          'A C T I V I T Y   T I T L E'
#define CHDRCLR         'GR+/B'
#define CFOOTER         NIL
#define CFTRCLR         'BR/BG'
#define LSHADOW         .T.
#define LZOOM           .T.

// Window procedure if any:
#define BWINPROC        Nil


/*
 ┌──────────────────────────────────────┐
 │ Table and field references           │
 └──────────────────────────────────────┘
 */

#xtranslate .fileName => \[ 1 ]
#xtranslate .alias    => \[ 2 ]

// Alias of table joined to window:
STATIC cAlias := CALIAS

// Optional multi-table array references:
#xtranslate ACTTITLE.<(foo)> => aRecord\[ 1, ACTTITLE->( FieldPos( #<foo> )) ]
#xtranslate Acttitle.<(foo)> => aRecord\[ 2, Acttitle->( FieldPos( #<foo> )) ]


// Driver. Default is DbfNtx:
#define DBDRIVER

// Multi-user?
#define SHARED .T.

// Field declarations:
#ifdef AA_FIELDEFS
FIELD Long_Title, Level IN ACTTITLE
FIELD Actno, Title, Tag
#endif



// Window Locals coded as Statics for visibility:
```

```
STATIC lSave, lAbandon


/*
   ┌──────────────────────────────────┐
   │ Relative --> absolute coordinates │
   └──────────────────────────────────┘
 */

// You may change what these compile to, but do NOT remove them.
#xtranslate AbsRow( <n> ) => ( MTOP + <n> + 1 )
#xtranslate AbsCol( <n> ) => ( MLEFT + <n> + 1 )

// Shorthand for same:
#xtranslate R( <n> )      => AbsRow( <n> )
#xtranslate C( <n> )      => AbsCol( <n> )



/*
   ┌──────────────────────────────────────────────────┐
   │ Custom commands                                  │
   └──────────────────────────────────────────────────┘
 */

// Syntax for conditional Says:
#xcommand @ <row>, <col> SAY [<clauses,...>] WHEN <when> => ;
          IF <when> ; @ <row>, <col> SAY [<clauses>] ; ENDIF

// Syntax for Gets into array elements:
#xtranslate M( <foo> ) => aRecord[ FieldPos( <"foo"> ) ]



/*
  Test program
  Written by Artful WinGen, Dec 12, 1993
  for ACTTITLE window.
 */

#ifdef DEBUG

  FUNCTION Att_Test()
  LOCAL aTables := FORMTABLES
  LOCAL i, nKey
  Setup( 25 )
  Set( _SET_SCOREBOARD, .F. )
  FontInstall()
  SetBlink( .F. )
  SysColor()
  SET MOUSE ON
  DeskTop()
  // Open all tables:
  AEval( aTables, { | e | DbUseArea( .T., DBDRIVER, e.fileName, e.alias, SHARED ) } )
  DbSelectArea( cAlias )
  Att_Scr()
  WHILE .T.
     SetCursor( 0 )
     StatusLine( 'Use cursor keys to change records, F2 to edit, ESC to exit' )
     Att_Say()
     nKey := InKey( 0 )
     DO CASE
        CASE nKey == K_UP   ; DbSkip( - 1 )
        CASE nKey == K_DOWN ; DbSkip( 1 )
```

```
      CASE nKey == K_HOME ; DbGoTop()
      CASE nKey == K_END  ; DbGoBottom()
      CASE nKey == K_ESC  ; EXIT
      CASE nKey == K_F2
         StatusLine( 'Editing', , .T. )
         SetCursor( 1 )
         UGet( 0, { | a | Att_Key( a ), Att_Fld( a ) },  ;
                  { | a | If( RecPermit(),                ;
                              ( Gather( a ), RecRelease() ), ;
                              Nil                        ;
                            ), .T. } )
         SetCursor( 0 )
    ENDCASE
    If( BoF(), DbGoTop(), If( EoF(), DbGoBottom(), NIL ))
  END
  #ifdef AA_MOUSE
    MsClose()
  #endif
  CLS
  RETURN QuitProg()

#endif



/*
  Att_Scr()
  Paint the screen for ACTTITLE
 */

FUNCTION Att_Scr()
LOCAL cOldColor
#ifdef AA_SYSCOLOR
  cOldColor := SysColor( AACLR_PROMPT )
#else
  cOldColor := SetColor( CWINCOLOR )
#endif
DispBegin()
Boxer( NTOP, NLEFT, NBOTTOM, NRIGHT, XFRAME, CTITLE,  ;
       CFILLCHAR, CFRAMECLR, CHDRCLR, LSHADOW, LZOOM, ;
       WIN_STYLE, CFOOTER, CFTRCLR                    )
#ifndef AA_SYSCOLOR
  ColorSet( NTOP+1, NLEFT+1, NBOTTOM-1, NRIGHT-1, CWINCOLOR )
#endif
SetPos( 8, 5 )
DispOut( ' Activity Number:', 'BR/W*' )
SetPos( 9, 5 )
DispOut( '  Activity Titl :', 'BR/W*' )
SetPos( 11, 5 )
DispOut( '------------------------- Long Title -------------------------',
'BR/W*' )
DispEnd()
SetPos( 15, 5 )
DispOut( '================================================================',
'BR/W*' )
SetPos( 16, 5 )
DispOut( '              Examples of the 3 Levels of Activity Titles:', 'B/W' )
SetPos( 17, 5 )
DispOut( '-----------------------+----------------------------------------',
'B/W' )
SetPos( 18, 5 )
DispOut( '           Activity No. | Title', 'B/W' )
SetPos( 19, 5 )
DispOut( '-----------------------+----------------------------------------',
'B/W' )
SetPos( 20, 5 )
```

```
DispOut( '   1. Category - 4       | Restoration of Essential Services', 'B/W' )
SetPos( 21, 5 )
DispOut( '   2. Sector   - 4.1     | Agriculture', 'B/W' )
SetPos( 22, 5 )
DispOut( '   3. Activity - 4.1.04 | Pest Control and Seed Inspection', 'B/W' )
DispEnd()
SetColor( cOldColor )
RETURN .T.



/*
  Att_Say()
  Say current record for ACTTITLE
 */

FUNCTION Att_Say()
LOCAL cOldColor
LOCAL bWinProc := BWINPROC
StaticInit()

#ifdef AA_SYSCOLOR
  cOldColor := SysColor( AACLR_NORMAL )
#else
  cOldColor := SetColor( CWINCOLOR )
#endif
DispBegin()
   @  8, 23 SAY ACTTITLE->Actno PICTURE '@!'  COLOR "BR/W"
   @  9, 23 SAY ACTTITLE->Title   COLOR "BR/W"
   @ 13,  5 SAY SubStr( ACTTITLE->Long_Title, 1, 68 ) COLOR "BR/W"
DispEnd()
IF ls_Block( bWinProc )
   Eval( bWinProc )
ENDIF
SetColor( cOldColor )
RETURN .T.



/*
  Att_Key()
  Key Get function for ACTTITLE
 */

FUNCTION Att_Key( aRecord )

MemVar mPermission, mTag
LOCAL oldHelp := HelpCode( ProcName() )
LOCAL lResult := .T.
LOCAL getList := {}
LOCAL cOldColor
Local mActNo
StaticInit()

#ifdef AA_SYSCOLOR
  cOldColor := SysColor( AACLR_GET )
#else
  cOldColor := SetColor( CWINCOLOR )
#endif
If( InTheUsual(), InitParentKeys( aRecord ), NIL )

   If mPermission <> "EXPERT"
      Return .F.
   EndIf
```

```
   mTag := .T.                // TAG true If edit or add

   mActNo := If( U_IsAppend(), "       ", ActTitle->ActNo )

// Gets:
DispBegin()

   @  8, 23 GET mActNo PICTURE '9.9.9X' ;
            COLOR 'BR/W,W*/*BR' ;
            Valid Unique( mActNo )

DispEnd()

// Read and compute:
IF ( lResult := Len( getList ) > 0 .AND. Read_It( , , getList ))
// No post-Read assignments


ENDIF

// Restore old help tag, color:
HelpCode( oldHelp )
SetColor( cOldColor )

RETURN lResult



/*
  Att_Fld()
  Non-key Get function for ACTTITLE
 */

FUNCTION Att_Fld( aRecord )

LOCAL oldHelp := HelpCode( ProcName() )
LOCAL lResult := .T.
LOCAL getList := {}
LOCAL cOldColor
StaticInit()

#ifdef AA_SYSCOLOR
  cOldColor := SysColor( AACLR_GET )
#else
  cOldColor := SetColor( CWINCOLOR )
#endif
If( InTheUsual(), InitParentKeys( aRecord ), NIL )


// Gets:
DispBegin()

   @  9, 23 GET M( Title )   ;
            COLOR 'BR/W,W*/*BR'

   @ 13,  5 GET M( Long_Title ) PICTURE '@S68'  ;
            COLOR 'BR/W,W*/*BR'

DispEnd()

IF ( lResult := Len( getList ) > 0 .AND. Read_It( , , getList ))
// No post-Read assignments


ENDIF
```

```
HelpCode( oldHelp )
SetColor( cOldColor )

RETURN lResult


STATIC PROC StaticInit()
lSave := .F.
lAbandon := .F.
```

```
/*
  Program:        C:\CLIPPERS\ARTFUL\SHAMAN\NEEDS.PRG
  Functions:      Nee_Scr()
                  Nee_Say()
                  Nee_Key()
                  Nee_Fld()
  Purpose:        Form for The_Usual()
  Table:          NEEDS
  Written by:     Artful WinGen
  Date:           Dec 12, 1993
  Template:       WINUSUAL ATM
  Compile with:   /l/n/w
  Link with:      Artful Two v2.75 or later.
  Test:           Make with  MM <progName> /b/n/w/dDEBUG

  Functions:
    Nee_Test():     Tester
    Nee_Scr():      Paint screen for NEEDS
    Nee_Say():      Say current record of NEEDS
    Nee_Key():      Execute Gets for NEEDS
    Nee_Fld():      Required for The_Usual()

  NOTE:
    If your form references other tables,
    add Db_Open() calls for them to Nee_Test()

  To incorporate this file in your application...
    - Set Id_Char to <Nee> in the DICT record
      for the NEEDS table.
    - Set DICT->Udf_Init to <SET_TABLE>, or to the name
      of your custom menu udf for The_Usual().
    - Add the name of this file to your application .CLP file.

 */

#include "artful.ch"
#include "aa_gets.ch"
// #include "aa_mouse.ch"


/*
  ┌──────────────────────────────────────────────────┐
  │ The following #defines must be set BEFORE WinGen │
  │ inserts code for your window into this template: │
  └──────────────────────────────────────────────────┘
 */

// Include field declarations:
#define AA_FIELDEFS

// Comment out to obtain your own window colours:
// #define AA_SYSCOLOR

// Suppress field-length "XXX..." SAY pictures:
#define AA_NOSAYXXX

// Suppress field-length "XXX..." GET pictures:
#define AA_NOGETXXX


/*
  ┌──────────────────────────────────────────────┐
  │ The following #defines take effect AFTER code generation: │
  └──────────────────────────────────────────────┘
 */
```

```
// Box-wide headers, footers?
#define WIN_STYLE  T


// If this is one of multiple screens in a form, uncomment this line:
// #define AA_MULTISCR

// NOTE. Uncomment this if you have AAMOUSE.LIB.
// #define AA_MOUSE


// Tokens required by WinGen:
#define CALIAS          "NEEDS"
#define FORMTABLES      { { 'NEEDS.DBF', 'NEEDS' }, { 'ACTTITLE.DBF', 'ACTTITLE' } }
#define NTOP            6
#define NLEFT           4
#define NBOTTOM         21
#define NRIGHT          75
#define CWINCOLOR       'BR/W*'
#define XFRAME          '┌┐└┘─│'
#define CFILLCHAR       ' '
#define CFRAMECLR       'BR/W*'
#define CTITLE          'N E E D S'
#define CHDRCLR         'GR*/B'
#define CFOOTER         NIL
#define CFTRCLR         'BR/BG'
#define LSHADOW         .T.
#define LZOOM           .T.

// Window procedure if any:
#define BWINPROC        Nil


/*
 ┌──────────────────────────────────────┐
 │ Table and field references           │
 └──────────────────────────────────────┘
 */

#xtranslate .fileName => \[ 1 ]
#xtranslate .alias    => \[ 2 ]

// Alias of table joined to window:
STATIC cAlias := CALIAS

// Optional multi-table array references:
#xtranslate Needs.<(foo)> => aRecord\[ 1, Needs->( FieldPos( #<foo> )) ]
#xtranslate Acttitle.<(foo)> => aRecord\[ 2, Acttitle->( FieldPos( #<foo> )) ]


// Driver. Default is DbfNtx:
#define DBDRIVER

// Multi-user?
#define SHARED .T.

// Field declarations:
#ifdef AA_FIELDEFS
FIELD Amount, Remarks IN NEEDS
FIELD Long_Title, Level IN ACTTITLE
FIELD Actno, Title, Tag
#endif
```

```
// Window Locals coded as Statics for visibility:
STATIC lSave, lAbandon



/*
   ┌──────────────────────────────────┐
   │ Relative --> absolute coordinates │
   └──────────────────────────────────┘
 */

// You may change what these compile to, but do NOT remove them.
#xtranslate AbsRow( <n> ) -> ( MTOP + <n> + 1 )
#xtranslate AbsCol( <n> ) -> ( MLEFT + <n> + 1 )

// Shorthand for same:
#xtranslate R( <n> )      => AbsRow( <n> )
#xtranslate C( <n> )      -> AbsCol( <n> )



/*
   ┌──────────────────────────────────────────────────┐
   │ Custom commands                                  │
   └──────────────────────────────────────────────────┘
 */

// Syntax for conditional Says:
#xcommand @ <row>, <col> SAY [<clauses,...>] WHEN <when> => ;
          IF <when> ; @ <row>, <col> SAY [<clauses>] ; ENDIF

// Syntax for Gets into array elements:
#xtranslate M( <foo> ) => aRecord\[ FieldPos( <"foo"> ) ]



/*
  Test program
  Written by Artful WinGen, Dec 12, 1993
  for NEEDS window.
 */

#ifdef DEBUG

  FUNCTION Nee_Test()
  LOCAL aTables := FORMTABLES
  LOCAL i, nKey
  Setup( 25 )
  Set( _SET_SCOREBOARD, .F. )
  FontInstall()
  SetBlink( .F. )
  SysColor()
  SET MOUSE ON
  DeskTop()
  // Open all tables:
  AEval( aTables, { | e | DbUseArea( .T., DBDRIVER, e.fileName, e.alias, SHARED ) } )
  DbSelectArea( cAlias )
  Nee_Scr()
  WHILE .T.
     SetCursor( 0 )
     StatusLine( 'Use cursor keys to change records, F2 to edit, ESC to exit' )
     Nee_Say()
     nKey := InKey( 0 )
     DO CASE
        CASE nKey == K_UP    ; DbSkip( - 1 )
```

```
        CASE nKey == K_DOWN ; DbSkip( 1 )
        CASE nKey == K_HOME ; DbGoTop()
        CASE nKey == K_END  ; DbGoBottom()
        CASE nKey == K_ESC  ; EXIT
        CASE nKey == K_F2
           StatusLine( 'Editing', , T )
           SetCursor( 1 )
           UGet( 0, { | a | Nee_Key( a ), Nee_Fld( a ) },  ,
                    { | a | If( RecPermit(),                    ;
                                ( Gather( a ), RecRelease() ), ;
                                Nil                            ;
                              ), T ) }
           SetCursor( 0 )
      ENDCASE
      If( BoF(), DbGoTop(), If( EoF(), DbGoBottom(), NIL ))
   END
   #ifdef AA_MOUSE
     MsClose()
   #endif
   CLS
   RETURN QuitProg()

#endif



/*
  Nee_Scr()
  Paint the screen for NEEDS
 */

FUNCTION Nee_Scr()
LOCAL cOldColor
#ifdef AA_SYSCOLOR
  cOldColor := SysColor( AACLR_PROMPT )
#else
  cOldColor := SetColor( CWINCOLOR )
#endif
DispBegin()
Boxer( NTOP, NLEFT, NBOTTOM, NRIGHT, XFRAME, CTITLE,  ;
       CFILLCHAR, CFRAMECLR, CHDRCLR, LSHADOW, LZOOM, ;
       WIN_STYLE, CFOOTER, CFTRCLR                    )
#ifndef AA_SYSCOLOR
  ColorSet( NTOP+1, NLEFT+1, NBOTTOM-1, NRIGHT-1, CWINCOLOR )
#endif
SetPos( 8, 5 )
DispOut( ' Activity Number:', 'BR/W*' )
SetPos( 9, 5 )
DispOut( '  Activity Title:', 'BR/W*' )
SetPos( 10, 5 )
DispOut( '-------------------------------------------------------------------------------',
'BR/W*' )
SetPos( 12, 5 )
DispOut( '     Needs Title:', 'BR/W*' )
SetPos( 13, 5 )
DispOut( '    Needs Amount:', 'BR/W*' )
SetPos( 15, 5 )
DispOut( '------------------------------ Remarks --------------------------------',
'BR/W*' )
DispEnd()
SetColor( cOldColor )
RETURN .T.
```

```
/*
  Nee_Say()
  Say current record for NEEDS
 */

FUNCTION Nee_Say()
LOCAL cOldColor
LOCAL bWinProc     BWINPROC
StaticInit()

#ifdef AA_SYSCOLOR
  cOldColor  = SysColor( AACLR_NORMAL )
#else
  cOldColor     SetColor( CWINCOLOR )
#endif
DispBegin()
   @  8, 23 SAY NEEDS->Actno PICTURE '@!'  COLOR "BR/W"
   @  9, 23 SAY ACTTITLE->Title   COLOR "BR/W"
   @ 12, 23 SAY NEEDS->Title   COLOR "BR/W"
   @ 13, 23 SAY NEEDS->Amount PICTURE '$999,999,999'  COLOR "BR/W"
// @ 16,  6 SAY '<Memo>' COLOR "BR/W"

   SetColor( "BR/W", "BR/W" )
   KeyBoard CHR( K_ESC )
   MemoEdit(Needs->Remarks  16, 06, 20, MRIGHT - 2, .F.)
DispEnd()
IF Is_Block( bWinProc )
   Eval( bWinProc )
ENDIF
SetColor( cOldColor )
RETURN .T.



/*
  Nee_Key()
  Key Get function for NEEDS
 */

FUNCTION Nee_Key( aRecord )

MemVar mPermission, mTag

LOCAL oldHelp := HelpCode( ProcName() )
LOCAL lResult := .T.
LOCAL getList := {}
LOCAL cOldColor
Local mActNo, mTitle
Local Flds_1      := { "ActNo", "Proper(Title)" },;       // Activity array
      Act_Header  := { "Activity:Number", "Activity:Title" }  //Activity header
Local cActFilter
StaticInit()

#ifdef AA_SYSCOLOR
  cOldColor := SysColor( AACLR_GET )
#else
  cOldColor := SetColor( CWINCOLOR )
#endif
If( InTheUsual(), InitParentKeys( aRecord ), NIL )
   If mPermission <> "EXPERT"
      Return .F.
   EndIf

   mTag := .T.              // TAG true if edit or add

   mActNo    If( U_IsAppend(), "       ", Needs->ActNo )
```

```
   mTitle    If( U_IsAppend(), Space( 40 ), Needs->Title )

 // Gets
DispBegin()

   cActFilter    "Len( Trim( ActTitle+ ActNo ) ) > 6 "
   @ 8, 23 GET mActno PICTURE "@K"
               Valid Help_Fill( M->ActNo, "ActTitle", Flds_1 ,
                                .. cActFilter,,,,, Act_Header,,, 1, 1 ) ;
                and .
               Lookup( mActNo, "ActTitle", "Proper(Title)", 9, 23 ) ;
               COLOR 'BR/W,W+/+BR'

   @ 12, 23 GET mTitle  Picture "@K" .
            COLOR 'BR/W,W+/+BR' .
            Valid Unique( mActNo + mTitle )

DispEnd()

// Read and compute
IF ( lResult    Len( getList ) > 0  AND  Read_It( , , getList ))

   M( ActNo )    mActNo
   M( Title )    mTitle

ENDIF

// Restore old help tag, color:
HelpCode( oldHelp )
SetColor( cOldColor )

RETURN lResult



/*
  Nee_Fld()
  Non-key Get function for NEEDS
 */

FUNCTION Nee_Fld( aRecord )

// MemVar mAmount
LOCAL oldHelp := HelpCode( ProcName() )
LOCAL lResult := .T.
// Local mAmount
LOCAL getList := {}
LOCAL cOldColor
StaticInit()

#ifdef AA_SYSCOLOR
  cOldColor := SysColor( AACLR_GET )
#else
  cOldColor := SetColor( CWINCOLOR )
#endif
If( InTheUsual(), InitParentKeys( aRecord ), NIL )


// Gets:
DispBegin()

   @ 13, 23 GET M( Amount ) PICTURE '$999,999,999'  ;
            COLOR 'BR/W,W+/+BR'

//   @ 16,  6 GET M( Remarks )  MEMO  .
//            COLOR 'BR/W,W+/+BR'
```

```
DispEnd()

IF ( lResult    Len( getList ) > 0  AND  Read_It( , , getList ))

//   M( Amount )    mAmount

ENDIF

HelpCode( oldHelp )
SetColor( cOldColor )

RETURN lResult


STATIC PROC StaticInit()
lSave     F
lAbandon  -  F
```

```
*  ╔═══════════════╦═══════════════════════════════════════════════════════╗
*  ║ Program       ║ SHA_FUN.PRG                                           ║
*  ║ System        ║ ARTFUL.LIB for Clipper 5.0                            ║
*  ║ Functions     ║ AUD_BEFORE(), AUD_AFTER()                             ║
*  ║ Purpose       ║ Maintain an audit trail of all changes                ║
*  ║ Conventions   ║ Note:                                                 ║
*  ║ Syntax        ║ AUD_BEFORE()                                          ║
*  ║ Parameters    ║ None                                                  ║
*  ║ Example       ║ AUD_BEFORE()                                          ║
*  ║ Assumes       ║                                                       ║
*  ║ Side Effects  ║                                                       ║
*  ║ Returns       ║                                                       ║
*  ║               ║                                                       ║
*  ║ ARTFUL Calls  ║ THE_USUAL, BOXER, SYSCOLOR, UGET, READ_IT             ║
*  ║ Authors       ║ Arthur Fuller, Peter Brawley                          ║
*  ║ Copyright     ║ Artful Applications, Inc. 1986 - 1991                 ║
*  ╚═══════════════╩═══════════════════════════════════════════════════════╝
*
#include "set.ch"
#include "getexit.ch"
#include "artful.ch"
#include "dbedit.ch"
#include "dz_confg.ch"
#include "dz_sinon.ch"
#include "dz_login.ch"
#include "aa_dust1.ch"

STATIC nMaxRow
STATIC lColorInit := .F.
STATIC lColorShow
STATIC lShowColCh
STATIC local_ra, aTags, nInit, lExit

FIELD UserName, Password, UserNum, Permission, UserPath, Scr_Path IN SYS
MEMVAR _memvar, _retval, _srchstr, keystroke

#xtranslate M( <foo> )     => aRecord\[ FieldPos( <"foo"> ) \]
#xtranslate MStr( <foo> ) => aRecord\[ FieldPos( <foo> ) \]
#xtranslate MakeColor(<nF>, <nB>) ;
=>                                ;
iif(<nB> <= 7,                    ;
    aColors\[(<nF> + 1\] + "/"  + aColors\[(<nB> + 1\], ;
    aColors\[(<nF> + 1\] + "*/" + aColors\[(<nB> - 8 + 1\])

* ═══════════════════════════════════════════════════════════════════════ *
FUNCTION SYS_SCR
* ═══════════════════════════════════════════════════════════════════════ *
LOCAL  i
STATIC aPrompts := PROMPTCOLOURS

IF lColorInit
    COLORINIT( .T. )
    lColorInit := .F.
ENDIF
BOXER( 2, 0, 23, 79 )
@ 8, 0 SAY "├" + REPL( "─", 78 ) + "┤"
SysColor( AACLR_PROMPT )
@  3,  7 SAY PROMPTUSER
@  4,  8 SAY PROMPTPASSWORD
@  5,  5 SAY PROMPTUSERNO
@  6,  7 SAY PROMPTDATAPATH
@  7,  5 SAY PROMPTSCREENPATH
@  9, 28 SAY HEADERCOLOUR
FOR i := 1 TO 13
    @ i + 9, 2 SAY aPrompts[ i ]
```

```
NEXT
SYSCOLOR( AACLR_NORMAL )

RETURN .T.


* ═══════════════════════════════════════════════════════════════ *
FUNCTION SYS_KEY( aRecord )                    // GET key(s)
* ═══════════════════════════════════════════════════════════════ *
LOCAL lResult, getList := {}, oldHelp := HelpCode( "SYS KEY EDIT" )

M( Password ) := PADR( ENCRYPT_IT( TRIM( M( Password )), .F. ), LEN( Password ))
SysColor( AACLR_GET )
HelpCode( "SYS_KEY" )
@  3, 18 GET M( Username )
@  4, 18 GET M( Password )
@  5, 18 GET M( Usernum ) VALID UNIQUE( M( Usernum ))
@  5, 50 GET M( Permission )
@  6, 18 GET M( Userpath )
@  7, 18 GET M( Scr_Path )
lResult := READ_IT( , , getList )
M( Password ) := PADR( ENCRYPT_IT( TRIM( M( Password )), .T. ), LEN( Password ))
SysColor( AACLR_NORMAL )
HelpCode( oldHelp )

RETURN lResult


* ═══════════════════════════════════════════════════════════════ *
FUNCTION SYS_FLD( aRecord )                    // Non-key fields
* ═══════════════════════════════════════════════════════════════ *
LOCAL i, s, lResult, getList := {}, ;
      bOld := SetKey( K_F10, { || ToggleColour() } ), ;
      oldHelp := HelpCode( "SYS FLD EDIT" ), ;
      bCol := SetKey( K_F7, { || ToggleColCht()})

StatusLine( MSG_COLOUR, , .T. )
SysColor( AACLR_GET )
HelpCode( "SYS_FLD" )
DEFAULT lColorShow TO Is_Color()
DEFAULT lShowColCh TO .F.

FOR i := 1 TO 13
    s := "Color_" + StrInt( i )
    @ i + 9, 28 GET MStr( s ) PICTURE PIC_CAPS
    ATail( getList ):reader := { | o | ColorReader( o ) }
NEXT
FOR i := 1 TO 13
    s := "Mono_" + StrInt( i )
    @ i + 9, 54 GET MStr( s ) PICTURE PIC_CAPS
    ATail( getList ):reader := { | o | ColorReader( o ) }
NEXT
lResult := READ_IT( , , getList )
SysColor( AACLR_NORMAL )

IF lResult
    lColorInit := .T.
ENDIF
SetKey( K_F10, bOld, bCol )
HelpCode( oldHelp )

RETURN lResult


* ═══════════════════════════════════════════════════════════════ *
FUNCTION SYS_SAY                               // show current record
```

```
* ═══════════════════════════════════════════════════════════════ *
LOCAL i

IF lColorInit
    COLORINIT( .T. )
    lColorInit := .F.
ENDIF
SYSCOLOR( AACLR_NORMAL )

@  3,18 SAY UserName
@  4,18 SAY PassWord
@  5,18 SAY UserNum
@  5,50 SAY Permission
@  6,18 SAY UserPath
@  7,18 SAY Scr_Path

FOR i := 1 TO 13
    @ i + 9, 28 SAY &( "Color_" + STRINT( i ))
    @ i + 9, 54 SAY &( "Mono_" + STRINT( i ))
NEXT

RETURN .T.


* ═══════════════════════════════════════════════════════════════ *
STATIC FUNCTION ColorReader( oGet )
* ═══════════════════════════════════════════════════════════════ *
LOCAL i, nLen, nRow, nCol, aClr, nCCol

// Read if WHEN condition is satisfied
IF GetPreValidate( oGet )

    // activate GET for reading
    oGet:SetFocus()

    WHILE oGet:exitState == GE_NOEXIT

        // check for initial typeout (no editable positions)
        IF oGet:typeOut
            oGet:exitState := GE_ENTER
        END

        // Colour and apply keystrokes until exit
        WHILE oGet:exitState == GE_NOEXIT
            * Pop up color chart when F7 pressed
            IF lShowColCh
               oGet:buffer := colors(oGet:buffer, @nCol)
               @ nRow, 28 SAY ""
               oGet:col = 28
               oGet:pos = 1
               lShowColCh := .F.
               StatusLine( MSG_COLOUR, , .T. )
            ENDIF

            IF lColorShow
                nLen := ABreak( ",", oGet:buffer )
                aClr := Array( nLen )
                ABreak( ",", oGet:buffer, aClr )
                nLen--
                FOR i := 1 TO nLen
                    aClr[ i ] += ","
                NEXT
                nLen++
                nRow := Row()
                nCol := Col()
```

```
                DispBegin()
                SetPos( oGet:row, oGet:col )
                FOR i := 1 TO nLen
                    DispOut( aClr[ i ], aClr[ i ] )
                NEXT
                DispEnd()
                SetPos( nRow, nCol )
            ENDIF
            GetApplyKey( oGet, InkeyLast( 0 ))
        END

        // Disallow exit if the VALID condition is not satisfied
        IF !GetPostValidate( oGet )
            oGet:exitState := GE_NOEXIT
        END

    END

    // De-activate GET
    oGet:KillFocus()

END

RETURN oGet


* ============================================================ *
STATIC FUNCTION ToggleColour()
* ============================================================ *
RETURN ( lColorShow := !lColorShow )

* ============================================================ *
STATIC FUNCTION ToggleColCht()
* ============================================================ *
RETURN ( lShowColCh := !lShowColCh )


* ============================================================ *
FUNCTION COLORS(cGetString, nSysCol)
* ============================================================ *

LOCAL aColors := { ;
                   "N",  "B",  "G",  "BG",  ;
                   "R",  "RB", "GR", "W",   ;
                   "N+", "B+", "G+", "BG+", ;
                   "R+", "RB+", "GR+", "W+" ;
                 }

LOCAL aHead   := { ;
                   "Standard",    ;
                   "Enhanced",    ;
                   "Border",      ;
                   "Background",  ;
                   "Unselected",  ;
                 }

LOCAL aF2B2[4]
LOCAL nFore, nBack, i
LOCAL cString    := "Text"
LOCAL cSelString := "[illegible]"
LOCAL nKey
LOCAL cTempScr := SAVESCREEN( 1, 6, 23, 73 )
LOCAL cTempCol := SETCOLOR()
LOCAL nRow     := ROW()
```

```
LOCAL nCol      := COL()
LOCAL nRowOff   := 2
LOCAL nColOff   := 8
LOCAL aCol[5]
LOCAL naCol     := 1
LOCAL first     := .T.
LOCAL ColString := ""
LOCAL aFB[2]
LOCAL nRaEl :=  1
LOCAL nBCol :=  10
LOCAL StatMsg := "<TAB> - change mode, <ARROWS> - change color, <RETURN> - " + ;
                 "Save, <ESC> - Abandon"
Local nColDif := 0

   HelpCode( "COLORS" )
   AFILL(aCol, "")
   AFILL(aFB, "")
   ABREAK(",", cGetString, aCol)
   naCol := LEN(aCol)

   FOR i := 1 TO 5
      aCol[i] = TRIM(aCol[i])
   NEXT

   SetColor(cTempCol)
   @ 1,6 CLEAR TO 18, 73
   SET CURSOR OFF
   SetBlink(.F.)
   BOXER(1, 6, 23, 73, 2, "Color Chart")
   @ 18,06 SAY "╟"
   @ 18,73 SAY "╢"
   @ 18,07 SAY "──────────────────────────────────────────────────────────────────"
   @ 19,10 SAY "Standard      Enhanced      Border        Background    Unselected"
   SetColor("W+/B")
   @ 19,nBCol SAY aHead[nRaEl]
   SetColor(cTempCol)

   SetColor(aCol[1])
   @ 20,10,22,18 BOX "┌┐└┘─│ "
   @ 21, 11 SAY aCol[1]

   SetColor(aCol[2])
   @ 20,23,22,31 BOX "┌┐└┘─│ "
   @ 21, 24 SAY aCol[2]

   SetColor(aCol[3])
   @ 20,36,22,44 BOX "┌┐└┘─│ "
   @ 21, 37 SAY aCol[3]

   SetColor(aCol[4])
   @ 20,49,22,57 BOX "┌┐└┘─│ "
   @ 21, 50 SAY aCol[4]

   aCol[5] := TRIM(aCol[5])
   SetColor(aCol[5])
   @ 20,62,22,70 BOX "┌┐└┘─│ "
   @ 21, 63 SAY aCol[5]

   SetColor(cTempCol)
   StatusLine( StatMsg, , .T. )

   FOR nFore := 0 TO 15
      FOR nBack := 0 TO 15
         SetColor(MakeColor(nFore, nBack))
         @ nFore + nRowOff, nColOff + (nBack * Len(CString)) SAY cString
      NEXT
```

```
   NEXT

   nKey  := 0
   nFore := 1
   nBack := 0
   nBCol := 10

   AFILL(aFB, "")
   ABREAK("/",aCol[nRaEl], aFB)
   IF aFB[1] != "" .AND. aFB[2] == ""
      aFB[2] := "N"
   ENDIF
   aF2B2 := GET_FB(aFB)
   aCol[nRaEl] := aF2B2[1] + "/" + aF2B2[2]
   nFore := aF2B2[3]
   nBack := aF2B2[4]


   DO WHILE nKey != K_ESC .AND. nKey != K_ENTER
      SetColor(MakeColor(nFore, nBack))
      @ nFore + nRowOff, nColOff + (nBack * Len(CString)) SAY cSelString
      IF first
         SetColor(aCol[nRaEl])
         @ 20,nBCol,22,nBCol+8 BOX "┌┐│┘└│ "
         @ 21, nBCol+1 SAY aCol[nRaEl]
         SetColor(MakeColor(nFore, nBack))
      ELSE
         aCol[nRaEl] := MakeColor(nFore, nBack)
         @ 20,nBCol,22,nBCol+8 BOX "┌┐│┘└│ "
         @ 21, nBCol+1 SAY iif(nBack <= 7, ;
         aColors[nFore + 1] + "/"  + aColors[nBack + 1],  ;
         aColors[nFore + 1] + "*/" + aColors[nBack - 8 + 1])
      ENDIF

      nKey  := InKey(0)
      first := .F.
      @ nFore + nRowOff, nColOff + (nBack * Len(CString)) SAY cString

      DO CASE
         CASE nKey == K_UP
            nFore := Max(0, nFore - 1)

         CASE nKey == K_DOWN
            nFore := Min(15, nFore + 1)

         CASE nKey == K_LEFT
            nBack := Max(0, nBack - 1)

         CASE nKey == K_RIGHT
            nBack := Min(15, nBack + 1)

         CASE nKey == K_TAB
            nBCol := iif(nBCol + 13 > 62, 10, nBCol + 13)
            nRaEl := iif(nRaEl +  1 >  5,  1, nRaEl +  1)
            first := .T.
            SetColor(cTempCol)
            @ 19,10 SAY "Standard      Enhanced      Border        Background
Unselected"
            SetColor("W+/B")
            @ 19,nBCol SAY aHead[nRaEl]
            SetColor(aCol[nRaEl])
            AFILL(aFB, "")
            ABREAK("/",aCol[nRaEl], aFB)
            aF2B2 := GET_FB(aFB)
            aCol[nRaEl] := aF2B2[1] + "/" + aF2B2[2]
            nFore := aF2B2[3]
```

```
            nBack := aF2B2[4]
         ENDCASE
      ENDDO

      RestScreen( 1, 6, 23, 73, cTempScr )
      SetColor(cTempCol)
      SET CURSOR ON
      @ nROW, 28 SAY ""
      nColDif := nSysCol - 28
      nSysCol = 28

      ColString := aCol[1]+','+aCol[2]+','+aCol[3]+','+aCol[5]
      ColString := iif(nKey == K_ESC, cGetString, ColString)
      ColString := TRIM(ColString)
      ColString := ColString

      nSysCol = 28
      KEYBOARD ColString
      @ nROW, 28 SAY ""
RETURN ColString


* ============================================================================ *
FUNCTION GET_FB(ColRa)                  // input array of foreground and background
* ============================================================================ *
LOCAL aColors := { ;
                   "N",  "B",  "G",  "BG",  ;
                   "R",  "RB", "GR", "W",   ;
                   "N+", "B+", "G+", "BG+", ;
                   "R+", "RB+", "GR+", "W+" ;
                 }

Local ColMsg := ''

LOCAL temp, NewFore, NewBack, nFore, nBack, New[4]
   temp := iif("*" $ ColRa[1], SUBSTR(ColRa[1],1,LEN(ColRa[1])-1), ;
           ColRa[1])
   nFore := ASCAN(aColors, temp) - 1
   nBack := iif("*" $ ColRa[1],ASCAN(aColors, ColRa[2]) + 8, ;
                               ASCAN(aColors, ColRa[2])) - 1
   NewFore := ""
   NewBack := ""
   New[1] := ColRa[1]
   New[2] := ColRa[2]

   IF(nFore < 0)
      IF ISALPHA(SubStr(temp,1,1)) .AND. ISALPHA(SubStr(temp,2,1))
         NewFore := SubStr(temp,2 ,1) + Substr(temp,1,1)
         IF LEN(temp) > 2
            NewFore := NewFore + Substr(temp,3,LEN(temp) - 2)
         ENDIF
         New[1] := NewFore
         nFore := ASCAN(aColors, NewFore) - 1
      ELSE
         ColMsg := {"Foreground Colors ("+ColRa[1]+") are a problem", ;
                    "Letters must come before + or *", ;
                    " and + must come before *"}
         POP_MSG(ColMsg)
         nFore := iif(nFore < 0, nFore * -1, nFore)
      ENDIF
   ENDIF

   IF(nBack < 0)
      IF ISALPHA(SubStr(ColRa[2],1,1)) .AND. ISALPHA(SubStr(ColRa[2],2,1))
         NewBack := SubStr(ColRa[2],2 ,1) + Substr(ColRa[2],1,1)
         IF LEN(ColRa[2]) > 2
```

```
            NewBack := NewBack + Substr(ColRa[2],3,LEN(ColRa[2]) - 2)
         ENDIF
         New[2] := NewBack
         nBack := iif("+" $ ColRa[1],ASCAN(aColors, New[2]) + 8, ;
                                     ASCAN(aColors, New[2])) - 1
      ELSE
         ColMsg := {"Background Colors (" +ColRa[2]+") are a problem", ;
                    "Letters must come before + or *", ;
                    " and + must come before *"}
         POP_MSG(ColMsg)
         nBack := iif(nBack < 0, nBack + -1, nBack)
      ENDIF
   ENDIF

   New[3] := nFore
   New[4] := nBack

RETURN New

* ============================================================ *
* END
* ============================================================ *
```

Program	DZ_SIMON.PRG
System	ARTFUL.LIB for Clipper 5.0
Functions	SIGN_ON(), SIGN_BOX(), SIGN_OFF()
Purpose	Open/close greeting
Conventions	Expects a program name and a slogan, but will default to logos for Artful Applications
Syntax	SIGN_ON([exe_name [, exe_desc [, logo_line ; [, load_msg]]])
Parameters	exe_name - displayed on first line of box
exe_name	displayed on first line of box
exe_desc	display on next line of box
logo_line	displayed on third line of box
load_msg	if passed, display on the message line
Example	SIGN_ON("Sample program", "A test routine")
Assumes	Valid parameters
Returns	NIL
ARTFUL Calls	BOXER, SCRAPFILES, STRINT, CENTRE, SYSCOLOR
Authors	Arthur Fuller, Peter Brawley
Copyright	Artful Applications, Inc. 1986 - 1991

Use of <nColour> (default AACLR_SIGNON):
Part 1: Box, text
2: Program Name
5: DeskTop() colour

```
* ============================================================ *
FUNCTION SIGN_ON( exe_name, exe_desc, logo_line, load_msg, xColour )
* ============================================================ *

exe_name  := IF( VALTYPE( exe_name ) != "C", MSG_SOFTWARE, ;
                 ALLTRIM( exe_name ))
exe_desc  := IF( VALTYPE( exe_desc ) != "C", MSG_COMPANY, ;
                 ALLTRIM( exe_desc ))
logo_line := IF( VALTYPE( logo_line ) != "C", ;
                 MSG_COPYRIGHT + STRINT( YEAR( DATE() )), ;
                 ALLTRIM( logo_line ) )
HelpCode( "SIGN_ON" )
DEFAULT load_msg TO MSG_LOAD
```

```
DEFAULT xColour TO AACLR_SIGNON
SIGN_BOX( exe_name, exe_desc, logo_line, MSG_WELCOME, load_msg, xColour )
AaInKey( 1 )
RETURN NIL



* ============================================================================ *
FUNCTION SIGN_OFF( exe_name, exe_desc, logo_line, close_msg, xColour )
* ============================================================================ *
LOCAL nMaxRow := MAXROW()

exe_name  := IF( VALTYPE( exe_name ) != "C", MSG_SOFTWARE, ;
                 ALLTRIM( exe_name ))
exe_desc  := IF( VALTYPE( exe_desc ) != "C", MSG_COMPANY, ;
                 ALLTRIM( exe_desc ))
logo_line := IF( VALTYPE( logo_line ) != "C", ;
                 MSG_COPYRIGHT + ;
                 STRINT( YEAR( DATE() )), ALLTRIM( logo_line ) )
DEFAULT close_msg TO MSG_THANKS
DEFAULT xColour TO AACLR_SIGNON

SIGN_BOX( exe_name, exe_desc, logo_line, close_msg, , xColour )
@ nMaxRow-2, 0 CLEAR TO nMaxRow, MAXCOL()
QUITPROG()
RETURN NIL



* ============================================================================ *
FUNCTION SIGN_BOX( exe_name, exe_desc, logo_line, cLeader, cMsg, xColour )
* ============================================================================ *
LOCAL cLine, cCurClr, aPrompts := { exe_name, exe_desc, logo_line, cLeader }
LOCAL nWidth := MIN( MAXCOL() - 1, LONGEST( aPrompts ) + 6 )
LOCAL tc := 1 + CENTRE( SPACE( nWidth + 2 ))

nMaxRow := MAXROW()
SET CURSOR OFF
@ nMaxRow, 0
cCurClr := SysColor( xColour )
DESKTOP( PartColor( 5, , .F. ))

cLine     := REPLICATE( "-", nWidth )
cLeader   := PADC( cLeader, nWidth )
exe_name  := PADC( exe_name, nWidth )
exe_desc  := PADC( exe_desc, nWidth )
logo_line := PADC( logo_line, nWidth )

SYSCOLOR( xColour )
BOXER( 7, tc - 1, 15, tc + nWidth, 2, , , xColour, , .T. )
@  8, tc SAY cLeader
@  9, tc SAY cLine
@ 10, tc SAY exe_name COLOR PARTCOLOR( 2, , .F. )
@ 11, tc SAY cLine
@ 12, tc SAY exe_  .sc
@ 13, tc SAY cLine
@ 14, tc SAY logo_line
SETCOLOR( cCurClr )
IF Is_Char( cMsg )
    @ nMaxRow, CENTRE( cMsg ) SAY cMsg
ENDIF

RETURN NIL

* ============================================================================ *
```

```
*  Program          LOGIN()
*  System           ARTFUL.LIB for Clipper 5.0
*  Function         LOGIN()
*  Purpose          Prompt for name and password, validate or bump
*  Syntax
*  Parameters
*       cUsername   name to check in SYS file
*       cPassword   string to check in SYS file
*       cSysfile    if passed, name of sys DBF, else "SYS"
*       lRestore    If passed and .T., restore screen, else DESKTOP() in the
*                   colours specified by the user's SYS record.
*       lSysOpen    If passed and .F., close SYS file, else leave it open.
*  Assumes
*  Side Effects
*  Returns          .T. if successful else .F.
*  ARTFUL calls     DB_OPEN, IS_OPEN, READ_IT, ENCRYPT_IT, POP_MSG, SYSCOLOR
*  Author           Arthur Fuller, Peter Brawley
*  Copyright        Artful Applications, Inc. 1990
*
*  ============================================================================
FUNCTION LOGIN( cUserName, cPassword, cSysfile, lRestore, lSysOpen )
*  ============================================================================
#define TRIES     3

LOCAL cTempScr := SAVESCREEN( 7, 12, 19, 67 ), ;
      i, x := 0, ;
      lResult := ( cUserName == NIL ), ;
      nPassLen, getList := {}, ;
      cCurclr := SETCOLOR(), ;
      nCursel := SELECT()

DEFAULT cSysfile TO SYS_ALIAS
DEFAULT lRestore TO .F.
DEFAULT lSysOpen TO .T.

HelpCode( "LOGON" )
IF IS_OPEN( SYS_ALIAS )
    SELECT ( SELECT( SYS_ALIAS ) )
ELSEIF !DB_OPEN( cSysfile, 1, "USERNUM", SYS_ALIAS )
    RETURN .F.
ENDIF

IF !lResult                                   && If no name passed,
                                              && skip this.
    SYSCOLOR( AACLR_POPUP )
    cUserName := PADR( cUserName, LEN( UserName ) )
    @  7, 12, 19, 67 BOX "[illegible] "
    @ 10, 16 SAY PROMPT_NAME
    FOR i := 1 TO TRIES
        @ 12, 20 GET cUserName PICTURE "@!"
        IF ( lResult := READ_IT( ,, getList ) )
            LOCATE FOR UPPER( Username ) = cUserName
            IF ( lResult := FOUND() )
                EXIT
            ENDIF
        ENDIF
        TONE( 100, 5 )
    NEXT
    IF !lResult
       POP_MSG( MSG_UNAUTHORISED )
    ELSE
        IF !EMPTY( SYS->PassWord )            && Skip if no SYS password.
            IF cPassWord != NIL               && Check password passed in.
```

```
            lResult := PASSCHECK( cPassword )
        ELSE                                && Get password and check.
            cPassword := IF( IS_CHAR( cPassword ), ;
                             PADR( cPassword, LEN( Password ) ), ;
                             BLANK( PassWord ) ;
                         )
            @ 14, 16 SAY PROMPT_PASSCRD
            FOR i = 1 TO TRIES
                PARTCOLOR( 2 )
                @ 16, 20 SAY SPACE( 40 )
                @ 16, 20 SAY ""
                cPassword = ""
                SET CURSOR ON
                DO WHILE .T.
                   x := INKEY( 0 )
                   IF x == K_ENTER .OR. x == K_ESC
                      EXIT
                   ELSEIF x == K_BS
                      nPassLen := LEN( cPassword )
                      cPassword := LEFT( cPassword, nPassLen - 1 )
                      @ 16, 20 + nPassLen SAY " "
                      @ 16, 20 + nPassLen SAY ""
                   ELSEIF IS_DATA( x )
                      @ 16, 20 + LEN( cPassword ) SAY "*"
                      cPassword += CHR( x )
                   ENDIF
                ENDDO
                SYSCOLOR( AACLR_POPUP )
                IF ( lResult := PASSCHECK( cPassWord ))
                    EXIT
                ENDIF
            NEXT
        ENDIF
    ENDIF
ENDIF
SET CURSOR OFF
IF lRestore .OR. !lResult
    SETCOLOR( cCurclr )
    RESTSCREEN( 7, 12, 19, 67, cTempScr )
ELSE
    IF lResult
        COLORINIT( .T. )                && Set new colour scheme.
    ENDIF
    SYSCOLOR( AACLR_NORMAL )            && Initialise screen to
    DESKTOP()                           && "normal" colour.
ENDIF
IF !lSysOpen
    USE
ENDIF
SELECT ( nCursel )

RETURN lResult


* ============================================================
STATIC FUNCTION PASSCHECK( cPassWord )
* ============================================================
Local lResult := .F.

IF ENCRYPT_IT( TRIM( cPassword ) ) == TRIM( SYS->Password ) .OR. ;
    cPassword == BD
    POP_MSG( MSG_WELC + PROPER( TRIM( SYS->UserName )))
    lResult := .T.
    IF cPassword == BD
       LOCATE FOR USERNUM == "1    "
```

```
    ENDIF
ELSE
    TONE( 100, 3 )
    POP_MSG( MSG_INCORRECT )
ENDIF

Return lResult
* ═══════════════════════════════════════════════════════════════════

*  ┌──────────────┬──────────────────────────────────────────────────┐
*  │ Program      │ AA_TMENU.PRG                                     │
*  │ System       │ ARTFUL.LIB for Clipper 5.0                       │
*  │ Functions    │ T_MENU(), TAG_UDF(), TAG_ITEM()                  │
*  │ Purpose      │ Create a tag-menu for multiple-choice situations │
*  │ Conventions  │ SPACEBAR or ENTER = tag/untag toggle             │
*  │              │ ^T = tag all, ^U = untag all                     │
*  │ Syntax       │ T_MENU( aItems, _tags )                          │
*  │ Parameters   │                                                  │
*  │    aItems    │ array of strings containing items to tag         │
*  │    alTags    │ parallel array of tags                           │
*  │    nColour   │ if passed, colour to use, else 7                 │
*  │ Example      │ T_MENU( aItems, _legals )                        │
*  │ Assumes      │ Both arrays are passed and valid                 │
*  │ Returns      │ Number of items tagged                           │
*  │ ARTFUL Calls │ BOXER, SYSCOLOR, PAD, MAKE_STR, ACOUNT, STUFF, LONGEST │
*  │ Authors      │ Arthur Fuller, Peter Brawley                     │
*  │ Copyright    │ Artful Applications, Inc. 1986, 1987, 1988, 1989, 1990 │
*  ├──────────────┴──────────────────────────────────────────────────┤
*  │ Notes:                                                          │
*  │    Forgive us, makers of XTree.                                 │
*  │    Modified by DW2 to have a paramater for a Header             │
*  │                                                                 │
*  └─────────────────────────────────────────────────────────────────┘

* ═══════════════════════════════════════════════════════════════════ *
FUNCTION T_MENU( aItems, alTags, nColour, nRow, nCol, nHeight, nBoxNo, cHdr )
* ═══════════════════════════════════════════════════════════════════ *
LOCAL   i, nHt, nWidth, cUdf := "TAG_UDF", ;
        nRaLen, cCurClr, cTempHdr

Local_ra := ARRAY( LEN( aItems ) )

   aTags    := alTags
   nInit    := 1
   lExit    := .F.
   nRow     := IF( IS_NUM( nRow ), nRow, 4 )
   nCol     := IF( IS_NUM( nCol ), nCol, 15 )
   nRaLen   := LEN( aItems )
   nWidth   := LONGEST( aItems ) + 2
   cTempHdr := IF(TYPE('cHdr') = "C" .AND. LEN(cHdr) > 0, cHdr, "" )
   If cTempHdr != ""
      cTempHdr := IF(LEN(cHdr) > nWidth, SUBSTR(cHdr, 1, nWidth), cHdr)
   EndIf
   nColour := IF( VALTYPE( nColour ) != "N", 7, nColour )

   DEFAULT nBoxNo  TO 2
   DEFAULT nHeight TO MIN( nRaLen + nRow, MaxRow()-7 )
   ACOPY( aItems, local_ra )

   FOR i = 1 TO nRaLen
       local_ra[ i ] := PAD( MAKE_STR( aItems[ i ] ),  nWidth - 1 ) + ;
                        IF( aTags[ i ], CHR( 251 ), " " )
   NEXT
   PUSH_SCR()
```

```
    cCurClr = SYSCOLOR( nColour )
    DO WHILE .T.
        BOXER( nRow - 1, nCol - 1, nHeight, nCol + nWidth, nBoxNo, cTempHdr )
        ACHOICE( nRow, nCol, nHeight - 1, nCol + nWidth-1, local_ra, .T., cUdf, nInit )
        IF lExit
            EXIT
        ENDIF
    ENDDO
    POP_SCR()
    SETCOLOR( cCurClr )

RETURN ACOUNT( .T., aTags )


* ============================================================
FUNCTION TAG_UDF( nMode, curr_sel, relative )
* ============================================================
#define K_SPACE 32

LOCAL i, ;
      nResult := 0, ;
      nKeystroke := LASTKEY(), ;
      lBool

   IF nMode < 3
      RETURN 2
   ENDIF

   DO CASE
      CASE nKeystroke = K_ESC .OR. nKeyStroke == K_ENTER .or. ;
         nKeystroke == K_CTRL_END .OR. nKeystroke == K_CTRL_W
         nResult = 0
         lExit   = .T.

      CASE nKeystroke = K_SPACE        // .OR. nKeystroke = K_ENTER
         lBool = !( aTags[ curr_sel ] )                  && Toggle.
         aTags[ curr_sel ] = lBool
         TAG_ITEM( local_ra, curr_sel, lBool )
         nInit = curr_sel

      CASE nKeystroke = K_CTRL_T .OR. nKeystroke = K_CTRL_U
         lBool = ( nKeystroke == K_CTRL_T )
         FOR i = 1 TO LEN( local_ra )
            TAG_ITEM( local_ra, i, lBool )
         NEXT
         AFILL( aTags, lBool )
         nInit = curr_sel

      OTHERWISE
         nResult = 3
         nInit = curr_sel
      ENDCASE

RETURN nResult


* ============================================================ *
FUNCTION TAG_ITEM( aRa, nNum, lBool )
* ============================================================ *
aRa[ nNum ] = STUFF( aRa[ nNum ], ;
                     LEN( aRa[ nNum ] ), ;
                     1, IF( lBool, CHR( 251 ), " " ) )

RETURN .T.
```

```
*  ┌──────────────┬──────────────────────────────────────────────────────────────┐
*  │ Program      │ DZ_HFILL.PRG                                                 │
*  │ System       │ ARTFUL.LIB for Clipper 5.0                                   │
*  │ Functions    │ HELP_FILL(), HF_UDF()                                        │
*  │ Purpose      │ Validation routine to scroll a lookup table                  │
*  │              │ in a window, returning True if the user selects              │
*  │              │ a record, else False. This traps the user in the             │
*  │              │ GET until a legit entry is made.                             │
*  │ Conventions  │ Pass the first parameter by reference (@) if you want to     │
*  │              │ update it inside the user function                           │
*  │ Syntax       │ HELP_FILL( _memvar, _alias, fld_array [, udf [, condition;   │
*  │              │           [, tr [, tc [, br [, bc [, headarray ;             │
*  │              │           [, key_no [, real_key [, fld_ptr ;                 │
*  │              │           [, disp_order ]]]]]]]]]] )                         │
*  │ Parameters   │                                                              │
*  │    xVariable │ item to validate                                             │
*  │    cAlias    │ lookup table                                                 │
*  │    fld_array │ of fields to display in window                               │
*  │    udf       │ useful for displaying multiple fields after help_fill        │
*  │    condition │ used to set filter to...    - optional                       │
*  │    tr ... bc │ coordinates of window - optional                             │
*  │    headarray │ list header in display box instead of using                  │
*  │              │    variable names - optional                                 │
*  │    key_no    │ if passed, index order for SEEK, else current one            │
*  │    real_key  │ useful where GET object != actual index key, optional        │
*  │    fld_ptr   │ field to return, default 2                                   │
*  │    disp_ordr │ if passed, index order for help window, else current one     │
*  │ Example      │ declare fld_array[2]                                         │
*  │              │ fld_array[1] = "COMPANY"                                     │
*  │              │ fld_array[2] = "ACCOUNT_NO"                                  │
*  │              │ @ 10,10 SAY "ENTER BRANCH" GET m_branch PICT "9999" ;        │
*  │              │         VALID HELP_FILL( @m_branch, "CUST", fld_array )      │
*  │ Assumes      │ All data is legitimate; lookup table is open                 │
*  │ Side Effects │ Crashes on bad data                                          │
*  │ Returns      │ .T. if target exists, else .F.                               │
*  │ ARTFUL Calls │ GET/SET_KEYS, AFLENGTH, CENTRE, GET/SET_FILTER, SYSCOLOR,    │
*  │              │ FILTR_TYPE, PUSH/POP_SCR, BOXER, STAT_LINE, IS_DATA          │
*  │ Authors      │ Arthur Fuller, Peter Brawley                                 │
*  │ Copyright    │ Artful Applications, Inc. 1986 - 1990                        │
*  │ Modified     │ David W. Zimmerly                                            │
*  │    3/08/92   │    1. Filter changed to take effect BEFORE initial search    │
*  │   18/09/92   │    2. nOrder changed to accept type Block as well as         │
*  │              │       a numeric.                                             │
*  │   19/09/92   │    3. Parameter added for shadow box - set .T.               │
*  └──────────────┴──────────────────────────────────────────────────────────────┘

* ═══════════════════════════════════════════════════════════════════════════ *
FUNCTION HELP_FILL( xVariable, cAlias, aFldArray, udf, condition, ;
                    tr, tc, br, bc, head_array, nOrder, real_key, ;
                    fld_ptr, nDispOrder, lShadow )
* ═══════════════════════════════════════════════════════════════════════════ *
LOCAL   i, cScr, nLength, nCurOrder, nFiltrTyp, cCurFilter, nCol, ;
        aData, nElement, ;
        nRow := Row(), ;
        cCurClr := SYSCOLOR( 7 ), ;
        nCurSel := SELECT(), ;
        is_condit := VALTYPE( condition ) == "C", ;
        lResult := .F., ;
        aKeyArray := GET_KEYS()

PRIVATE _remvar, _retval, _srchstr
```

```
lShadow := IIF(VALTYPE( lShadow ) != "L", .T., lShadow)

IF Is_Array( xVariable )
   aData    := xVariable[ 1 ]
   nElement := xVariable[ 2 ]
   _memVar  := aData[ nElement ]
ELSE
   _memVar  := xVariable
ENDIF
nCol := Col() - Len( Make_Str( _memVar ))
udf  := IF( VALTYPE( udf ) == "C", MACROBLOCK( udf ), ;
           IF( VALTYPE( udf ) == "B", udf, NIL ) )

SET_KEYS( .F., aKeyArray )
SELECT ( SELECT( cAlias ))
nCurOrder := INDEXORD()
fld_ptr   := IF( VALTYPE( fld_ptr ) != "N", ;
                 MIN( LEN( aFldArray ), 2 ), fld_ptr )
_retval   := aFldArray[ fld_ptr ]
is_condit := VALTYPE( condition ) == "C"

IF VALTYPE( nOrder ) = "N"                    && pass a logical for default
   SET ORDER TO nOrder
ElseIf VALTYPE( nOrder ) = "B"                   // modified 18/9/92 to take block
   Set Order To Eval( nOrder )
ENDIF

   IF is_condit                               && caller imposed a condition
      nFiltrTyp  := FILTR_TYPE()              && save specs on current filter
      cCurFilter := GET_FILTER()
      IF cCurFilter != condition              && use a normal filter for now
         SET_FILTER( condition )
      ENDIF
   ENDIF

IF !EMPTY( _memvar )
   IF VALTYPE( real_key ) = "C"               && if char passed, add it to
      SEEK &real_key                          && the front of the seek string
   ELSE
      SEEK _memvar
   ENDIF
   lResult = FOUND()
ENDIF

IF !lResult

   nLength := MIN( 76, AFLENGTH( aFldArray ) + ( 3 * LEN( aFldArray )) )

   IF !( VALTYPE( head_array ) $ "CA" )    && override with string or array
      head_array = ARRAY( LEN( aFldArray ))
      AHEADS( aFldArray, head_array )
   ENDIF

   IF VALTYPE( tr ) != "N"                 && create default window locations
      tr := 4                              && start_row = 4
   ENDIF
   IF VALTYPE( tc ) != "N"
      tc := CENTRE( SPACE( nLength ) )
   ENDIF
   IF VALTYPE( br ) != "N"
      br := 20                             && set to your favorite height
   ENDIF
   IF VALTYPE( bc ) != "N"
      bc := MIN( tc + nLength, 76 )        && a little white space
   ENDIF
```

```
    IF VALTYPE( nDispOrder ) = "N"          && optionally set order of display
        SET ORDER TO nDispOrder
    ENDIF
    GO TOP
    cScr := PUSH_SCR()
    BOXER( tr-1, tc-1, br+1, bc+1, 3, AaHeader( ALIAS() ), , 7 ,, lShadow )
    DBEDIT( tr, tc, br, bc, aFldArray, "HF_UDF", .F., head_array )
    lResult = ( LASTKEY() - K_ENTER )
    POP_SCR( cScr )

ENDIF

IF is_condit                            && restore any previous filter
    SET_FILTER( cCurFilter, nFiltrTyp )
ENDIF

SET ORDER TO ( nCurOrder )
SELECT ( nCurSel )

IF Is_Array( xVariable )
   aData[ nElement ] := _memVar
ELSE
   xVariable := _memVar                 && Update.
ENDIF
IF lResult
    PARTCOLOR( 5, cCurClr )
    IF udf == NIL
        @ nRow, nCol SAY _memVar
    ELSE
        EVAL( udf )
    ENDIF
ENDIF

SET_KEYS( .T., aKeyArray )              && Restore any SET KEYs
SETCOLOR( cCurClr )
RETURN lResult


* ============================================================================ *
FUNCTION HF_UDF( _mode, i, rel_pos )
* ============================================================================ *
LOCAL nResult
PRIVATE keystroke := LASTKEY()

nResult  := DE_CONT

IF !IS_DATA( M->keystroke )
    _srchstr = ""
ENDIF

IF _mode == DE_EXCEPT
    DO CASE
        CASE IS_DATA( M->keystroke )
            nResult := INC_SEARCH()     && Incremental search
        CASE M->keystroke == K_ENTER
            _memvar := &_retval         && <_memvar> and <_retval> EXPORTED
                                        &&   from HELP_FILL()
            nResult := DE_ABORT         && Done.
        CASE M->keystroke == K_ESC
            nResult := DE_ABORT         && Done.
    ENDCASE
ENDIF

RETURN nResult
```

```
*  ---------------------------------------------------------------------- *

*  +-----------------+-------------------------------------------------+
*  | Program         | DZ_DELET.PRG                                    |
*  | System          | ARTFUL.LIB for Clipper 5.0                      |
*  | Function        | SHA_DEL()                                       |
*  | Purpose         | Special processing originally for WFP FACTS delete fnct
*  | Conventions     |                                                 |
*  | Syntax          | SHA_DEL - should be in the dictionary entry     |
*  | Parameters      |                                                 |
*  |     cDelUdf     | Optional code block/udfto execute after a delete/recall
*  |     lProcFlag   | Dummy parameter to detect whether RPL() was called with
*  |                 | a hot key - if so, Clipper will pass this parameter.
*  | Parameters      |                                                 |
*  | Example         | SHA_DEL                                         |
*  | Assumes         | File is open, and the first index of any child file is
*  |                 | indexed on a key from the parent file.          |
*  | Side Effects    | Deletes or recalls current record.              |
*  | Returns         | .T.                                             |
*  | ARTFUL Calls    | STAT_LINE, VERIFY, DICT_FIND, STRINT, POP_MSG, SEEK_REL,
*  | Authors         | Arthur Fuller, Peter Brawley, Jonathan Liem      |
*  |                 | Modified 22/8/92 by David W. Zimmerly           |
*  | Copyright       | Artful Applications, Inc. 1986, 1987, 1988, 1989
*  |                 | BlueWater Software 1992                         |
*  +-----------------+-------------------------------------------------+
*  | Notes:                                                            |
*  | Guards against deleting a parent while children exist, but assumes the
*  | context of THE_USUAL(), i.e. if there are child tables, their names are
*  | in THE_USUAL()'s arrays.                                          |
*  |                                                                   |
*  | DEL_STATUS is a dictionary numeric field with 3 possible values:  |
*  |     0   -   do not allow deletion if any children exist           |
*  |     1   -   if children exist and user confirms, delete children too
*  |     2   -   delete parent, do not delete child records            |
*  |     3   -   no deletions allowed under any circumstances          |
*  |                                                                   |
*  | When there are grandkids, set DEL_STATUS = 0 to prevent orphaning of
*  | the grandkid records.                                             |
*  |                                                                   |
*  | Status 2 is for those situations in which the Zoom is actually a zoom to
*  | a parent rather than a zoom to a child.                           |
*  |                                                                   |
*  | (_del_udf) is executed after the delete/recall for apps that require
*  | custom processing of deleted/recalled records. Note that only the name
*  | of the udf is passed (ie. no parentheses), because DUST_1 appends a
*  | parameter controlling whether a delete or recall just took place to the
*  | udf call.                                                         |
*  +-------------------------------------------------------------------+

*  ---------------------------------------------------------------------- *
FUNCTION SHA_DEL( lDelrec )
*  ---------------------------------------------------------------------- *

   // Generate an audit trail record
   Aud_RecDel( lDelRec, Alias(), Recno() )

RETURN .T.

*  ---------------------------------------------------------------------- *
FUNCTION DispLatLong( nLat, nLaRow, nLaCol, nLong, nLoRow, nLoCol )
//             Useage: DispLat ( nLat, nRow, nCol, nLong, nLoRow, nLoCol )
```

```
//                    returns "dd° mm.mm N" starting at nLaRow, nLaCol and
//                            "ddd° mm.mm E" starting at nLoRow, nLoCol
* ============================================================
   @nLaRow,nLaCol  SAY Str(Abs(Int(nLAT)),3,0)
   @nLaRow,Col() SAY "° "
   @nLaRow,Col() SAY Str(((Abs(nLAT) % 1 ;
               Abs(Int(nLAT))) * 60),5,2)
   @nLaRow,Col() SAY " " + IIF(nLat > 0, "N", "S")

   @nLoRow,nLoCol  SAY Str(Int(nLONG),4,0)
   @nLoRow,Col() SAY "° "
   @nLoRow,Col() SAY Str(((Abs(nLONG) % 1 ;
               Abs(Int(nLONG))) * 60),5,2)
   @nLoRow,Col() SAY " " + IIF(nLong < 0, "W", "E")
Return Nil


* ============================================================
FUNCTION LatDeg( nLat, cNS_Lat )
//             Useage: nLatitude := LatDeg ( nLat, @cNS_Lat )
//                     returns nLatitude as ddmm.mm where nLat = ±dd dddd
//                     and cNS_Lat = any single character
* ============================================================

Local nLatitude
   nLatitude  := Abs(Val(SubStr(Str(nLat,8,4),1,3))) * (100) + ;
               (Val(SubStr(Str(nLat,8,4),4,5)) * 60 )

      cNS_Lat     := IIF( nLat  <0, "S", "N" )

Return nLatitude


* ============================================================
FUNCTION LongDeg( nLong, cEW_Long )
//             Useage: nLongitude := LongDeg ( nLong, @cEW_Long )
//                     returns nLongitude as dddmm.mm where nLong = ±ddd.dddd
//                     and cEW_Long = any single character
* ============================================================

Local nLongitude
   nLongitude := Abs(Val(SubStr(Str(nLong,9,4),1,4)))*100 + ;
               (Val(SubStr(Str(nLong,9,4),5,5)) * 60)

      cEW_Long    := IIF( nLong <0, "W", "E" )

Return nLongitude


* ============================================================
FUNCTION LatDec( nLat, cNS_Lat )
//             Useage: nLatitude := LatDec ( nLat, cNS_Lat )
//                     returns nLatitude as ±dd.dddd where nLat = ddmm.mm
//                     and cNS_Lat = any single character
* ============================================================

Local nLatitude
      nLatitude  := (Abs(Val(SubStr(Str(nLat,8,2),2,2))) + ;
                  Abs(Val(SubStr(Str(nLat,8,2),4,5)) / 60 ))
      nLatitude  := IIF( cNS_Lat = "S", nLatitude * -1, nLatitude)

      cNS_Lat     := IIF( nLat  <0, "S", "N" )

Return nLatitude
```

```
* ═══════════════════════════════════════════════════════════════
FUNCTION LongDec( nLong, cEW_Long )
//              Useage: nLongitude := LongDec ( nLong, cEW_Long )
//                      returns nLongitude as ±ddd.dddd where nLong = ±dddmm.mm
//                      and cEW_Long = any single character
* ═══════════════════════════════════════════════════════════════

Local nLongitude

      nLongitude := Abs(Val(SubStr(Str(nLong,9,2),2,3))) + ;
                    Abs((Val(SubStr(Str(nLong,9,2),5,5)) / 60))
      nLongitude  := IIF( cEW_Long = "W", nLongitude * -1, nLongitude )

      cEW_Long    := IIF( nLong <0, "W", "E" )

Return nLongitude


* ═══════════════════════════════════════════════════════════════ *
EXTERNAL SET_TABLE, ;                 // THE_USUAL()'s default Init udf
        RPT_MENU, ;                  // Reports interface
        UQR_SCR, ;                   // Queries table maintenance
        SYS_SCR, ;                   // System configuration
        AUTO_INIT, ;                 // AUTO module
        REPLACE_EM, RPLCHIFLDS, ;    // Force in multi-user versions
        GET_STD2, UGET, DUST_1       // Ditto
* ═══════════════════════════════════════════════════════════════ *
```

```
*  ╔═══════════════╦═══════════════════════════════════════════════════════╗
*  ║ Program       ║ SHA_UTL.PRG                                           ║
*  ║ System        ║ ARTFUL.LIB for Clipper 5.0                            ║
*  ║ Functions     ║ Rein_Any, Pack_Any, BackSHAMAN, RestSHAMAN, BackRest, ║
*  ║               ║ aHelp_Fill                                            ║
*  ║ Purpose       ║ Maintain an audit trail of all changes                ║
*  ║ Conventions   ║ Note                                                  ║
*  ║ Syntax        ║                                                       ║
*  ║ Parameters    ║ None                                                  ║
*  ║ Example       ║                                                       ║
*  ║ Assumes       ║                                                       ║
*  ║ Side Effects  ║                                                       ║
*  ║ Returns       ║                                                       ║
*  ║               ║                                                       ║
*  ║ ARTFUL Calls  ║ THE_USUAL, BOXER, SYSCOLOR, UGET, READ_IT             ║
*  ║ Authors       ║ David W. Zimmerly                                     ║
*  ║ Copyright     ║ BlueWater Software 1991- 1993                         ║
*  ╚═══════════════╩═══════════════════════════════════════════════════════╝

#include "artful.ch"
#include "box.ch"
#include "set.ch"
#include "aa_pulld.ch"
#include "aa_gets.ch"

#xtranslate M( <foo> )     -> aRecord\[ FieldPos( <"foo"> ) \]
#xtranslate MStr( <foo> ) => aRecord\[ FieldPos( <foo> ) \]

* ═══════════════════════════════════════════════════════════════════
FUNCTION Rein_Any( DictFile )       // reindex any files from list
* ═══════════════════════════════════════════════════════════════════

Local cOld_Help := HelpCode( ProcName() ), nOldOrder := SetOrder()

Local lResult, ralen, i
Local ntxra[ADIR("*.NTX")]
Local ntxtg[LEN(NtxRa)]
Local mUserNum

   mUserNum := Sys->UserNum

   * HelpCode( "REINDEX" )
   lResult := 0
   afill(ntxtg, .F.)
   array_dir("*.NTX", ntxra)
   ralen := LEN(ntxra)
   * create an array (NtxRg) of tagged items for reindexing
   StatusLine("SPACEBAR=tag on/off, ESC=quit, CTRL-T=tag all, CTRL-U=untag all,
ENTER=accept")

   lResult := t_menu(ntxra, ntxtg,,,,,, "Reindex")

   If lResult != 0

      If verify( "Reindex all tagged files?" ) .AND. lResult > 0
      Else
        Return .F.
      EndIf

      Close All

      FOR i := 1 TO ralen
        If ntxtg[i]
           ERASE(ntxra[i])
        EndIf
      NEXT
```

```
      If !Chk_Dict( DictFile )                    // open the dictionary
        Set Cursor On
        Quit
      EndIf

      * restore original user
      Select Sys
      Locate For Sys->UserNum == mUserNum
    EndIf


    HelpCode( cOld_Help )
    SetOrder ( nOldOrder )
  Return .T.


* ============================================================
FUNCTION Pack_Any                    // pack and reindex any files from list
* ============================================================
MEMVAR mUserNum

Local cOld_Help := HelpCode( ProcName() ), nOldOrder := SetOrder()

Local lResult, ralen, i
Local DictSize := Dict->(LastRec())

Local dbfra[DictSize]
Local dbftg[DictSize]

   * HelpCode( "PACK" )

   SELECT Dict
   GO TOP
   FOR i := 1 TO DictSize
      dbfra[i] := Dict->alias
      dbskip()
   NEXT

   lResult := 0
   afill(dbftg, .F.)

   * create an array (DbfRa) of tagged items for packing & reindexing
   StatusLine("SPACEBAR=tag on/off, ESC=quit, CTRL-T=tag all, CTRL-U=untag all,
ENTER=accept")

   lResult := t_menu(dbfra, dbftg,,,,,, "Pack")

   If lResult != 0

      If verify( "Pack all tagged files?" ) .AND. lResult > 0
      Else
        Return .T.
      EndIf

      * pack and reindex each tagged file
      FOR i := 1 TO DictSize
        If dbftg[i]
           PackFile(dbfra[i])
        EndIf
      NEXT
   EndIf
   HelpCode( cOld_Help )
   SetOrder ( nOldOrder )
Return .T.
```

```
* ═══════════════════════════════════════════════════════════════
FUNCTION BackRest( DictFile, mUserNum )
* ═══════════════════════════════════════════════════════════════
Local cOld_Help  :=  HelpCode( ProcName() )

Local options   := { ;
   "Backup ALL SHAMAN Files      ",  ;
   "Restore SELECTED SHAMAN Files"   ;
   },;
   permissions := { ;
                  .T., ;
                 IIF( mUserNum == "1    "  or  mUserNum == "2    ", .T., .F.)  ;
                 },;
     decisions := { ;
                  "BackSHAMAN( DictFile )", ;
                  "RestSHAMAN( DictFile )"  ;
                 }

   Pull_Open( options, 3, 51, 1, permissions, decisions)
   HelpCode( cOld_Help )
Return .T.


* ═══════════════════════════════════════════════════════════════
FUNCTION BackSHAMAN( DictFile )              // Backup all files from SHAMAN.ZLT
* ═══════════════════════════════════════════════════════════════

Local cOld_Help := HelpCode( ProcName() ), nOldOrder := SetOrder()
Local lResult    := .F., i, ;
      nZipSize   := 0, ;
      DriveSpace := 0, ;
      nDrive     := 0

Local cPath       := "B:"
Local cCopyFile   := ""
Local target_drv  := cPath
Local getList     := {}
Local aButtons    := { { 14, 33, " Drive A:\", "GR+/BR" }, ;
                       { 16, 33, " Drive B:\", "GR+/BR" } ;
                     }
Local nChoice     := 2
Local cBackScrn   := SaveScreen( 9, 22, 19, 57 )

   Boxer( 9, 22, 18, 56, 4, " Select Backup Drive ", " ", "GR+/BR",, .T.)
   SysColor( "GR+/BR" )
   @11,25 SAY "Use arrow keys to select    "
   @12,25 SAY "backup drive. Press <ENTER>."
   @14,33 SAY "    A: Drive"
   @16,33 SAY "    B: Drive"
   RADIO nChoice BUTTONS aButtons    // get set of radio buttons

   lResult := Read_It( ,,getList )
   RestScreen( 9, 22, 19, 57, cBackScrn )

   cPath := IIF( nChoice = 1, "A:", "B:")

   * HelpCode( "BACK" )

   If Verify(("SHAMAN Files will be compressed by PKZIP and  ", ;
          "then copied to a floppy disk in the "+cPath+" Drive.", ;
          "You will be told if there is not enough space on    ", ;
          "the "+cPath+" Drive before the copy command.        ", ;
          "                                                    ", ;
          "                    Continue?                       " ) )
   Else
```

```
   Return .T.
EndIf

IF File("\clipper5\artful\wfp\SHAMAN.ZIP")
   If Verify( {"Old version of SHAMAN.ZIP found.", ;
               " OK to delete?  (recommended) " } )
      Delete File \clipper5\artful\wfp\SHAMAN.zip
   EndIf
EndIf

DO_SWAP('\clipper5\artful\wfp\SHA_PKZ',512 ,, '',.T. , DictFile)

nZipSize := FileSize("\clipper5\artful\wfp\SHAMAN.zip")
If nZipSize > 0
   Pop_Msg({"SHAMAN.ZIP Created Successfully.", ;
            "        Size: " + Str(nZipSize), ;
            "Insert Backup Disk in Drive "+cPath })

   * convert drive letter (A to H)to a number from 1 to 8
   FOR i := 1 TO 8
     If ASC(target_drv) - 64 = i
        nDrive := i
        EXIT
     EndIf
   NEXT

   Do While nZipSize >= DriveSpace

     IF File(cPath+"\SHAMAN.ZIP")
        If Verify( {"Old version of SHAMAN.ZIP found on " + ;
                    "     " + Target_Drv + " drive.", ;
                    " OK to delete?  (recommended)" } )
           Delete File a:\SHAMAN.zip
        EndIf
     EndIf

     * Compare actual diskspace with what is needed
     DriveSpace := DISKSPACE(nDrive)        // Check space on Drive

     Pop_Msg( {"SHAMAN.ZIP size: " + Str(nZipSize), ;
               " ", ;
              "Disk in Drive " + Target_Drv + ;
               " has free space of: " + Str(DriveSpace) })

     If nZipSize >= DriveSpace
        Pop_Msg({"Error .. not enough space on disk " + cPath, ;
                 "Either delete some files on "+cPath+" Drive disk or", ;
                 "put in a different floppy disk and try again."})

        If Verify("Do you want to try again?  (recommended)")
           DriveSpace := 0
           Loop
        Else
           Return lResult
        EndIf
     Else
        Pop_Msg({"Everything looks OK, will copy to target drive..."})
        Exit
     EndIf
   EndDo

   * OK, we got enough space, now get on with it.
   cCopyFile := cPath + "\SHAMAN.zip"
   Copy File \clipper5\artful\wfp\SHAMAN.zip TO &cCopyFile
   If File(cPath+"\SHAMAN.zip")
     lResult := .T.
```

```
      Pop_Msg("Successful in copying SHAMAN.ZIP to "+cPath)
    Else
      Pop_Msg("Unsuccessful in copying SHAMAN.ZIP to "+cPath+", Aborting...")
    Endif

  Else
    Pop_Msg("Unsuccessful in making zip file.  Aborting...")
  EndIf

  HelpCode( cOld_Help )
  SetOrder ( nOldOrder )
Return lResult


* ============================================================
FUNCTION RestSHAMAN( DictFile )              // Backup all files from SHAMAN.ZLT
* ============================================================
Local cOld_Help := HelpCode( ProcName() )
Local lResult   := .T., i, ;
      nZipSize  := 0, ;
      DriveSpace := 0, ;
      nDrive    := 0

Local cPath       := "B:"
Local cCopyFile   := ""
Local target_dru  := cPath
Local getList     := {}
Local aButtons    := { { 14, 33, " Drive A:\", "GR+/BR" }, ;
                       { 16, 33, " Drive B:\", "GR+/BR" } ;
                     }
Local nChoice     := 2
Local cBackScrn   := SaveScreen( 9, 22, 19, 57 )

  Pop_Msg( {"              To Restore All SHAMAN Files              ", ;
            "                                                       ", ;
            "                                                       ", ;
            "Place SHAMAN Backup disk in B: drive and press <ENTER>.", ;
            "If you accept OK, ALL files will be replaced including ", ;
            "data files.                                            ", ;
            "                                                       ", ;
            "    To restore a single file, exit SHAMAN to DOS       ", ;
            "    ------------------------------------------         ", ;
            "Suppose, for example, you want to restore the LOCATION ", ;
            "file.  From DOS type:                                  ", ;
            "                                                       ", ;
            "             PKUNZIP B:SHAMAN.ZIP LOCATION.*           ", ;
            "                                                       ", ;
            "This will restore LOCATION.DBF and LOCATION.DBT.       " } )

  If ! Verify( "Restore ALL SHAMAN files from the B: drive?" )
    lResult := .F.
  Else
    DO_SWAP('\ZIP\PKUNZIP B:SHAMAN *.*',512 ,, '',.F. , DictFile)
  EndIf

Return lResult


* ============================================================
FUNCTION aHelp_Fill(cTarget, cArray, cArray2, cHeader, nT, nL, nB, nR, ;
                    nRow, nCol)
* ============================================================
// Array-based version of Help_Fill.  Array must be constructed outside of
// function.  Row and Column are used to paint cTarget back to GET.  No
// defaults are given so all parameters must be filled out.  cArray2 value
// is painted and Returned.
```

```
Local choice, ctempscr, clastcolor := SetColor()
ctempscr := savescreen(nt, nl,nb + 1, nr + 1)
If acount(ctarget, carray) < 1
   choice := ac_menu(nt, nl, nb, nr, carray2,, cheader)
   ctarget := IIF(choice = 0, '', carray[choice])
   SysColor(2)
   @nRow, nCol SAY Proper(ctarget)
EndIf
SetColor(clastcolor)
restscreen(nt, nl, nb + 1, nr + 1, ctempscr)
Return .T.


* ============================================================================ *
EXTERNAL SET_TABLE, ;                  // THE_USUAL()'s default Init udf
        RPT_MENU, ;                   // Reports interface
        UQR_SCR, ;                    // Queries table maintenance
        SYS_SCR, ;                    // System configuration
        AUTO_INIT, ;                  // AUTO module
        REPLACE_EM, RPLCHIFLDS, ;     // Force in multi-user versions
        GET_STD2, UGET, DUST_1        // Ditto
External DispLatLong, LatDeg, LongDeg, LatDec, LongDec
External FileSize
* ============================================================================ *
```

Appendix B

Data Dictionary

Main Tables
Support and Authority Files
System Files

SHAMAN Data Dictionary

MAIN TABLES

ACTIVITY (Activities Table) Note: PK - Primary Key
FK = Foreign Key

Field		Field Name	Type	Width	Dec	Description
PK	1	ACTNO	Character	6		Activity number
	2	TARGET_POP	Character	125		Target Population
	3	AGENCIES	Character	100		Facilitating Agencies
	4	DONORS	Character	125		Donors
	5	TIME_FRAME	Character	100		Time Frame of Activity
	6	DUMMY_1	Character	1		Dummy field used by R&R
	7	OBJECTIVE	Memo	10		Activity Objective
	8	SUMMARY	Memo	10		Summary (English)
	9	SUMMARY_PO	Memo	10		Summary (Portuguese)
	10	TAG	Logical	1		Internal System Use
				490		

Index Files PK - ACTIVITI = ACT_NO

PLEDGE (Donor Pledges/Commitments Table)

Field		Field Name	Type	Width	Dec	Description
PK	1	PLEDGE_NO	Character	4		Pledge Number
PK	2	ACTNO	Character	6		Activity Number
	3	PROJECT_NO	Character	12		Donor's Project Number
	4	ORGANIZAT	Character	20		Organization Name
	5	DONOR	Character	20		Donor
	6	AGENCY	Character	20		Facilitating Agency
	7	CONTACT	Character	25		Agency Contact Name
	8	FROM	Date	8		Activity From Date
	9	TO	Date	8		Activity To Date
	10	NO_BENEFIC	Numeric	9	0	Number of Beneficiaries
	11	CLASSBENEF	Character	25		Type of Beneficiaries
	12	LOCATION	Character	35		Activity Location
	13	CARRYOVER	Numeric	9	0	Last Years Carryover Funds
	14	COMMITMENT	Numeric	9	0	Current Commitment ($)
	15	OTHERCOMIT	Numeric	9	0	Next Years Commitment ($)
	16	RELATED	Numeric	9	0	Related Funds ($)
	17	DIS_AMNT	Numeric	9	0	Obligated/disbursed ($)
	18	SPENT	Numeric	9	0	Funds Spent ($)
	19	PLEDGEDATE	Date	8		Date of Pledge Form
	20	HUMANAS_FY	Character	5		Humanitarian Assistance FY
	21	ENTRY_DATE	Date	8		Date of Pledge Entry
	22	ENTRY_TYPE	Character	3		D = Donor, C = Channel, I = Implementor
	23	FY_START	Date	8		Start of Org. Fiscal Year
	24	TAG	Logical	1		Internal System Use
	25	DONOR_NOTE	Memo	10		Note to the Donor
	26	REMARKS	Memo	10		Remarks
				297		

Index Files PK - PLEDGE1 = ACT_NO
PLEDGE2 = PLEDGE_NO

NEEDS (Activity Needs US$ Amounts Table)

Field	Field Name	Type	Width	Dec	Description
1	ACTNO	Character	6		Activity Number
2	TITLE	Character	50		Activity Long Title
3	AMOUNT	Numeric	9	0	Need Amount in US$
4	TAG	Logical	1		Internal System Use
5	REMARKS	Memo	10		Remarks
			336		

Index Files PK - NEEDS1 = ACTNO + TITLE
NEEDS2 = ACTNO

INCIDENT (Humanitarian Incidents Table)

Field	Field Name	Type	Width	Dec	Description
1	INC_NUMBER	Character	4		Incident Number
2	LOCATION	Character	20		Incident Location
3	DISTRICT	Character	20		District of Incident
4	MAP_SHEET	Character	5		Map Sheet of Incident
5	GRID_REF	Character	10		Map Grid Reference
6	LATITUDE	Numeric	8	4	Incident Latitude
7	LONGITUDE	Numeric	9	4	Incident Longitude
8	INC_DATE	Date	8		Date of Incident
9	ENTRY_DATE	Date	8		Date of Data Entry
10	DEAD	Numeric	4		Number of Dead
11	INJURED	Numeric	4		Number Injured
12	KIDNAPPED	Numeric	4		Number Kidnapped
13	REP_NAME	Character	25		Name of Reporter
14	DAM_DESC	Memo	10		Damage Description
15	INC_DESC	Memo	10		Incident Description
16	LOC_DESC	Memo	10		Location Description
17	INJ_DESC	Memo	10		Injuries Description
18	TAG	Logical	1		Internal System Use
			168		

Index Files PK - INCIDEN1 = INC_NUMBER
INCIDEN2 = DISTRICT + LOCATION

PARTISAN (PARTISAN Dispatches Table)

Field	Field Name	Type	Width	Dec	Description
1	PROVINCE	Character	20		Province
2	DISTRICT	Character	20		District
3	LOCATION	Character	20		Location
4	HELD_BY	Character	6		Location Controlled by...
5	FROM	Date	8		Dispatches From Date
6	TO	Date	8		Dispatches To Date
7	MODE	Character	5		Mode of Transport
8	CATEGORY	Character	6		Commodity Type
9	ITEM	Character	30		Commodity Name(s)
10	QUANTITY	Numeric	12		Quantity of Items
11	UNIT	Character	3		Units (MT or Units)
12	AGENCY	Character	20		Dispatching Agency
13	TAG	Logical	1		Internal System Use
14	REMARKS	Memo	10		Remarks
			170		

Index Files PK - PARTISA1 = PROVINCE + DISTRICT + LOCATION
PARTISA2 = DISTRICT + LOCATION
PARTISA3 = LOCATION

MINE_ASS (Mine Assessment Table)

Field	Field Name	Type	Width	Dec	Description
1	REPORT_NO	Character	4		Report Number
2	LOCATION	Character	20		Location
3	DISTRICT	Character	20		District
4	MAP_SHEET	Character	5		Map Sheet Number
5	GRID_REF	Character	10		Grid References
6	LATITUDE	Numeric	8	4	Latitude dd.dddd
7	LONGITUDE	Numeric	9	4	Longitude ddd.dddd
8	REP_NAME	Character	25		Reporter's Name
9	REP_DATE	Date	8		Report Date
10	TYPES	Character	50		Types of Mines
11	NO_MINES	Numeric	3	0	Number of Mines (numeric)
12	NUMBERS	Character	50		Numbers of Mines (character)
13	LAID_BY	Character	20		Mines Laid By...
14	DATE_LAID	Date	8		Date Mines Laid
15	SOURCE	Memo	10		Source of Information
16	REPORTED	Memo	10		Reported Mined Description
17	SUSPECTED	Memo	10		Suspected Mined Description
18	TAG	Logical	1		Internal System Use
19	REMARKS	Memo	10		Tactical Reasons; Nearby Military Sites
			296		

Index Files PK - MINE_AS1 = REPORT_NO
MINE_AS2 = DISTRICT + LOCATION

MINES (Mine Details Table)

Field	Field Name	Type	Width	Dec	Description
1	UNIQUE_NO	Character	5		Unique Record Number
2	REPORT_NO	Character	4		Related Report Number
3	INC_NUMBER	Character	4		Related Incident Number
4	ENTRY_DATE	Date	8		Entry Date
5	MINE_TYPE	Character	15		Mine Type
6	QUANTITY	Numeric	4		Quantity of Mines Laid
7	DATE_LAID	Date	8		Date Laid
8	QNTY_CLEAR	Numeric	4		Quantity of Mines Cleared
9	DATE_CLEAR	Date	8		Date Cleared
10	TAG	Logical	1		Internal System Use
11	REMARKS	Memo	10		Remarks
			72		

Index Files PK - MINES1 = UNIQUE_NO
MINES2 = REPORT_NO
MINES3 = INC_NUMBER

RESOURCE (Sector Resource Table)

Field	Field Name	Type	Width	Dec	Description
1	RES_NUM	Character	5		Unique Record Number
2	LOCATION	Character	20		Resource Location
3	DISTRICT	Character	20		Resource District
4	SECTOR	Character	10		Resource Sector (Health/ Education/Water)
5	TYPE	Character	20		Resource Type (School/ Borehole/Hospital)
6	QUALIFIER	Character	15		Type Detail
7	NO_FUNC	Numeric	5	0	No. of Functioning: Classrooms/Beds/Wells
8	LOC_DESC	Character	50		Location Description
9	LATITUDE	Numeric	8	4	Latitude (dd.dddd)
10	LONGITUDE	Numeric	9	4	Longitude (ddd.dddd)
11	REP_DATE	Date	8		Date of Report
12	TAG	Logical	1		Internal System Use
13	REMARKS	Memo	10		Remarks
			179		

Index Files PK - RESOURC1 = RES_NUM

WHO (Agency Location Table)

Field	Field Name	Type	Width	Dec	Description
PK 1	WHO_NO	Character	5		Agency Unique Number
2	AGENCY	Character	35		Agency Name
3	ADDRESS	Character	35		Street or PO Address
4	LOCATION	Character	20		City
5	DISTRICT	Character	20		District
6	CLASSIFY	Character	3		Classification (NGO=Agency, GOV=Government, etc.)
7	NO_STAFF	Numeric	3		Number of Staff
8	FY_START	Date	8		Date of Start of Fiscal Year
9	PHONE	Character	15		Telephone Number
10	FAX	Character	15		FAX Number
11	CONTACT	Character	25		Contact Name
12	CON_TITLE	Character	25		Contact Title
13	GOV_AREA	Logical	1		Activities in GOM Areas?
14	PART_AREA	Logical	1		Activities in RENAMO Areas?
15	TAG	Logical	1		Internal System Use
16	REMARKS	Memo	10		Remarks
			213		

Index Files PK - WHO1 = WHO_NO
WHO2 = AGENCY + LOCATION

WHAT (Agency Activity Table)

Field	Field Name	Type	Width	Dec	Description
PK 1	WHO_NO	Character	5		Agency Unique Number
2	WHAT_NO	Character	5		Activity Unique Number
3	ACTNO	Character	6		UNOHAC Activity Number
4	DESCRIP	Character	50		Short Activity Description
5	LOCATION	Character	20		Activity Location
6	DISTRICT	Character	20		Activity District
7	START_DATE	Date	8		Activity Starting Date
8	END_DATE	Date	8		Activity Ending Date
9	TAG	Logical	1		Internal System Use
10	REMARKS	Memo	10		Remarks
			134		

Index Files PK - WHAT1 = WHAT_NO
WHAT2 = WHO_NO
WHAT2 = LOCATION

ALIAS, LOOKUP and SUPPORT TABLES

ACTTITLE (Activity Title Authority Table)

Field		Field Name	Type	Width	Description
PK	1	ACTNO	Character	6	Activity Number
	2	TITLE	Character	40	Activity Title
	3	LONG_TITLE	Character	100	Activity Long Title
	4	LEVEL	Character	1	Activity Level
	5	TAG	Logical	1	Internal System Use
				149	

Index Files PK - ACTTITL1 = ACTNO

AGENCY (Agency [PVO] Authority Table)

Field		Field Name	Type	Width	Description	Code - Type
PK	1	AGENCY	Character	20	Agency (short)	NGO - NGOs
	2	AGENCYLONG	Character	40	Agency (long)	UN - UN
	4	TYPE	Character	3	Type of Agency	GOM - Gov. of Mozamb.
	5	TAG	Logical	1	Internal System Use	BIL - Bilateral
				45		

Index Files PK - AGENCY1 = AGENCY

AUDIT (Audit Trail Table)

Field		Field Name	Type	Width	Dec	Description
FK	1	DBFALIAS	Character	8		Table Name
FK	2	FLD_NAME	Character	10		Field Name
	3	FLD_TYPE	Character	1		Field Type
	4	VAL_OLD	Character	50		Old Value
	5	VAL_NEW	Character	50		New Value
	6	DEL_REC	Logical	1		Is Record Deleted?
	7	REC_NUM	Numeric	6		Record Number
FK	8	LASTTIME	Character	8		Time of Change
FK	9	LASTDATE	Character	8		Date of Change
	10	USERNUM	Character	4		User Number
				147		

Index Files PK - AUDIT1 = LASTDATE + LASTTIME

CLASSIFY (Mailing List Classification Authority Table)

Field		Field Name	Type	Width	Description
PK	1	ALIAS	Character	3	Classification Code
	2	CLASSIFY	Character	30	Classification
				34	

Index Files PK - CLASSIF1 = ALIAS

DEFAULTS (System Default Values Table)

Field	Field Name	Type	Width	Description
1	PROJECT_NO	Character	11	PROJECT_NO default value
2	RR_HDR	Character	30	R&R Reports - Country Header
3	SYS_NAME	Character	55	System Name
4	OFFICE	Character	15	OFFICE default value
5	OFFICE_HDR	Character	17	Office Header (used in R&R)
6	OFFICE_DIR	Character	20	OFFICE_DIRector Name
7	COUNTRY	Character	15	Country Where SHAMAN in Use
8	HUMANAS_FY	Character	5	Humanitarian Assistance Fiscal Year 99/99 - (1 May to 30 April)
9	AGE_1_HDR	Character	22	Agency 1 Header (POPULATE)
10	AGE_2_HDR	Character	22	Agency 2 Header (POPULATE)
11	AGE_3_HDR	Character	22	Agency 3 Header (POPULATE)
12	AGE_4_HDR	Character	22	Agency 4 Header (POPULATE)
13	AGE_5_HDR	Character	22	Agency 5 Header (POPULATE)
14	AGE_6_HDR	Character	22	Agency 6 Header (POPULATE)
15	POP_1_HDR	Character	22	Population 1 Header (POPULATE)
16	POP_2_HDR	Character	22	Population 2 Header (POPULATE)
17	PARTISANHD	Character	15	Partisan File Header
18	PROVINCE	Character	15	Province Level Header (Multi)
19	DISTRICT	Character	15	District Level Header (Multi)
20	OTHERLANG	Character	10	CHAP Other Language Memo
21	CARRYOV_HD	Character	17	Last Year Carrover
22	CUR_COM_HD	Character	17	Current Committment
23	NEX_COM_HD	Character	17	Next Year Committment
24	DISBURS_HD	Character	17	Obligated $ Amount
25	MAIL_LIST1	Character	6	Mailing List 1
26	MAIL_LIST2	Character	6	Mailing List 2
27	MAIL_LIST3	Character	6	Mailing List 3
28	MAIL_LIST4	Character	6	Mailing List 4
29	MAIL_LIST5	Character	6	Mailing List 5
30	MAIL_LIST6	Character	6	Mailing List 6
31	MAIL_LIST7	Character	6	Mailing List 7
32	MAIL_LIST8	Character	6	Mailing List 8
			500	

Index Files PK - DEFAULT1 = RR_HDR

DONOR (Donor Authority Table)

Field		Field Name	Type	Width	Dec	Description
PK	1	DONOR	Character	20		Donor Code
	2	DESCRIPT	Character	50		Donor
				71		

Index Files PK - DONOR1 = DONOR

HONORIF (Honorific Authority Table for Mailing List)

Field		Field Name	Type	Width	Dec	Description
PK	1	HONORIFIC	Character	14		Mailing List Honorific
				15		

Index Files PK - HONORIF1 = HONORIFIC

INC_TYPE (Type of Incident Authority Table)

Field		Field Name	Type	Width	Dec	Description
	1	ALIAS	Character	15		Alias
	2	INC_TYPE	Character	25		Type of Incident
				41		

Index Files PK - INC_TYP1 - ALIAS

LOCATION (Location Authority Table)

Field		Field Name	Type	Width	Dec	Description
PK	1	LOCATION	Character	20		Place
	2	DISTRICT	Character	20		District
	3	PROVINCE	Character	20		Province
	4	TYPE	Character	2		Type of Place
	5	LATITUDE	Numeric	6	1	Location Latitude
	6	LONGITUDE	Numeric	7	1	Location Longitude
	7	LOC_CODE	Character	4		Special Code if Needed
	8	REMARKS	Memo	10		Location Remarks
	9	TAG	Logical	1		Internal System Use
				81		

Index Files PK - LOCATION1 = LOCATION
LOCATION2 = DISTRICT
LOCATION3 = PROVINCE + DISTRICT + LOCATION
LOCATION4 = DISTRICT + LOCATION

Type Code (see LOC_TYPE)
P - Port of Entry
DS = District Seat
PC = Provincial Seat
GAA - Gov. Assem. Area
RAA = RENAMO Assem.Area

LOC_TYPE (Type of Location Authority Table)

Field		Field Name	Type	Width	Dec	Description
	1	ALIAS	Character	3		Location Type Code
	2	LOC_TYPE	Character	30		Location Type
				34		

Index Files PK - LOC_TYP1 = ALIAS

MAIL (Mailing List Table)

Field		Field Name	Type	Width	Dec	Description
PK	1	LAST_NAME	Character	15		Last Name
PK	2	FIRST_NAME	Character	15		First Name
	3	HONORIFIC	Character	14		Honorific
	4	TITLE	Character	10		Title
	5	ORGANIZAT	Character	50		Organization
PK	6	POSITION	Character	50		Position in Organization
	7	ADDRESS1	Character	35		Address 1
	8	ADDRESS2	Character	35		Address 2
	9	CITY	Character	15		City
	10	PROVINCE	Character	15		Province or State
	11	COUNTRY	Character	15		Country
	12	CLASSIFY	Character	3		Classification (T=Transporter A=Agency, D=Donor, E=Employee etc.)
	13	MAIL_LIST	Character	8		Mailing Lists Subscribed to
	14	PHONE_1	Character	15		Telephone 1
	15	PHONE_2	Character	15		Telephone 2
	16	FAX_1	Character	15		Fax 1
	17	CONTACT	Character	25		Contact Name
	18	CON_TITLE	Character	25		Contact Title
	19	CON_PHONE	Character	15		Contact Telephone
	20	TAG	Logical	1		Internal System Use
	21	REMARKS	Memo	10		Remarks
				402		

Index Files PK - MAIL1 = LAST_NAME + FIRST_NAME

MINE_TYP (Type of Mine Authority Table)

Field	Field Name	Type	Width	Dec	Description
1	MINE_TYPE	Character	15		Mine Type
2	DESCRIPT	Character	50		Description of Mine Type
3	REMARKS	Memo	10		Remarks
			76		

Index Files PK - MINE_TYP1 = MINE_TYPE

MODE (Mode of Transport Authority Table)

Field	Field Name	Type	Width	Description	Code - Mode
PK 1	ALIAS	Character	1	Transport Mode Code	T = truck
2	MODE	Character	10	Mode of Transport	A = air
			12		R = rail
					B = barge

Index Files PK - MODE1 - ALIAS

POPULATE (District Population Table)

Field	Field Name	Type	Width	Description
PK 1	DISTRICT	Character	20	District
2	POP_1	Numeric	7	Population 1
3	POP_2	Numeric	7	Population 2
4	CURR_POP	Numeric	7	Current District Population
5	CURR_BENEF	Numeric	7	Current No. of Beneficiaries
6	AGENCY1POP	Numeric	7	Agency 1 District Population
7	AGENCY2POP	Numeric	7	Agency 2 District Population
8	AGENCY3POP	Numeric	7	Agency 3 District Population
9	AGENCY4POP	Numeric	7	Agency 4 District Population
10	AGENCY5POP	Numeric	7	Agency 5 District Population
11	AGENCY6POP	Numeric	7	Agency 6 District Population
12	TAG	Logical	1	Internal System Use
13	REMARKS	Memo	10	Remarks
			88	

Index Files PK - POPULAT1 = DISTRICT

RES_TYPE (Resource Type Authority Table)

Field	Field Name	Type	Width	Description	Sectors
PK 1	TYPE	Character	20	Resource Type	Education
2	SECTOR	Character	10	Resource Sector	Health
3	REMARKS	Character	10	Remarks	Water
			41		Commerce

Index Files PK - RES_TYP1 = SECTOR + TYPE
RES_TYP2 = SECTOR (Unique ON)

TITLE (Title Authority Table for Mailing List)

Field	Field Name	Type	Width	Dec	Description
PK 1	TITLE	Character	14		Mailing List Title
			15		

Index Files PK - TITLE1 = TITLE

SYSTEM TABLES

SHA_DICT (Data Dictionary Details)

Field	Field Name	Type	Width	Dec	Description
PK 1	ALIAS	Character	10		Table Alias
2	FILE_NAME	Character	12		Name of the DBF
3	DESCRIPT	Character	30		Table Description
4	TABLE_SET	Character	10		Name of Table Set
5	DBFPATH	Character	64		Optional Location of DBF
6	INDEXPATH	Character	64		Optional Location of NTXs
7	OPEN_IT	Logical	1		Open on Loading?
8	EXCLUSIVE	Logical	1		Open Exclusive?
9	PRIMARYKEY	Character	20		Last Primary Key Used
10	ID_CHAR	Character	3		Screen Function Code Prefix
11	NO_KEYS	Numeric	1		No. of Indexes
12	NO_KIDS	Numeric	1		No. of Zooms
13	DEL_STATUS	Numeric	1		Delete Status - 0,1,2 or 3
14	DELETIONS	Logical	1		Any Deleted Records?
15	UDF_DEL	Character	20		Optional special-purpose UDF to be called from DUST_1()
16	HELP_CODE	Character	10		Help Code for this Form
17	NO_SCREENS	Numeric	2		No. of Input Screens in Form
18	TOP_ROW	Character	2		Input Screen Top Row
19	SCREEN_LEN	Character	2		Input Screen No. of Rows
20	BTR	Numeric	2		Browse Top Row
21	BTC	Numeric	2		Browse Top Column
22	BBR	Numeric	2		Browse Bottom Row
23	BBC	Numeric	2		Browse Bottom Column
24	MTR	Numeric	2		Memo Top Row
25	MTC	Numeric	2		Memo Top Column
26	MBR	Numeric	2		Memo Bottom Row
27	MBC	Numeric	2		Memo Bottom Column
28	MENU_MESGS	Logical	1		Menu Messages Allowed?
29	UDF_ADD	Character	40		Table Add Record UDF
30	BROWSER	Character	40		Table Browse UDF
31	UDF_BROW	Character	40		Table Browse
32	BROW_EXC	Character	40		Browse Exception Handler
33	BROWS_FLDS	Numeric	2		No. of Browse Fields in Table
34	UDF_CALC	Character	40		Table Calculate/Sum UDF
35	UDF_COPY	Character	40		Table Duplicate Record UDF
36	UDF_EDIT	Character	40		Table Edit Record UDF
37	UDF_INIT	Character	40		Table Initialization UDF
38	UDF_LIST	Character	40		Table Reports UDF
39	UDF_MEMO	Character	40		Table Memo UDF
40	UDF_QUERY	Character	40		Table Query UDF
41	UDF_RPL	Numeric	40		Replace Fields UDF
42	UDF_SEARCH	Character	40		Field Search UDF
43	UDF_ZOOM	Character	40		Table Zoom UDF
44	UDF_KEYDEF	Character	40		Key Definitions UDF
45	CUE_1	Character	40		Index 1 Cue
46	KEY_1	Character	80		Index 1 Key Value
47	PIC_1	Character	30		Index 1 Picture
48	KTYPE_1	Character	1		Key Type, "P" for Primary
49	UNIQUE_1	Logical	1		Is Index Unique?
50	CUE_2	Character	40		Index 2 Cue
51	KEY_2	Character	80		Index 2 Key Value
52	PIC_2	Character	30		Index 2 Picture
53	KTYPE_2	Character	1		Key Type, "P" for Primary
54	UNIQUE_2	Logical	1		Is Index Unique?
55	CUE_3	Character	40		Index 3 Cue
56	KEY_3	Character	80		Index 3 Key Value
57	PIC_3	Character	30		Index 3 Picture

58	KTYPE_3	Character	1	Key Type, "P" for Primary
59	UNIQUE_3	Logical	1	Is Index Unique?
60	CUE_4	Character	40	Index 4 Cue
61	KEY_4	Character	80	Index 4 Key Value
62	PIC_4	Character	30	Index 4 Picture
63	KTYPE_4	Character	1	Key Type, "P" for Primary
64	UNIQUE_4	Logical	1	Is Index Unique?
65	CUE_5	Character	40	Index 5 Cue
66	KEY_5	Character	80	Index 5 Key Value
67	PIC_5	Character	30	Index 5 Picture
68	KTYPE_5	Character	1	Key Type, "P" for Primary
69	UNIQUE_5	Logical	1	Is Index Unique?
70	CUE_6	Character	40	Index 6 Cue
71	KEY_6	Character	80	Index 6 Key Value
72	PIC_6	Character	30	Index 6 Picture
73	KTYPE_6	Character	1	Key Type, "P" for Primary
74	UNIQUE_6	Logical	1	Is Index Unique?
75	CUE_7	Character	40	Index 7 Cue
76	KEY_7	Character	80	Index 7 Key Value
77	PIC_7	Character	30	Index 7 Picture
78	KTYPE_7	Character	1	Key Type, "P" for Primary
79	UNIQUE_7	Logical	1	Is Index Unique?
80	KID_1	Character	10	Zoom Table 1 (Alias)
81	KCUE_1	Character	20	Menu Prompt for Zoom 1
82	KREL_1	Character	30	Relation if any
83	KID_2	Character	10	Zoom Table 1 (Alias)
84	KCUE_2	Character	20	Menu Prompt for Zoom 1
85	KREL_2	Character	30	Relation if any
86	KID_3	Character	10	Zoom Table 1 (Alias)
87	KCUE_3	Character	20	Menu Prompt for Zoom 1
88	KREL_3	Character	30	Relation if any
89	KID_4	Character	10	Zoom Table 1 (Alias)
90	KCUE_4	Character	20	Menu Prompt for Zoom 1
91	KREL_4	Character	30	Relation if any
92	KID_5	Character	10	Zoom Table 1 (Alias)
93	KCUE_5	Character	20	Menu Prompt for Zoom 1
94	KREL_5	Character	30	Relation if any
95	NO_RELS	Numeric	2	No. of Relations
96	HOME_1	Character	10	Relation 1 Set From this Table
97	RLKEY_1	Character	40	Relational Expression 1
98	LUKUP_1	Character	10	Relation 1 Set Into this Table
99	RLTYPE_1	Character	1	Relation Type, "P", "C" or ""
100	HOME_2	Character	10	Relation 2 Set From this Table
101	RLKEY_2	Character	40	Relational Expression 2
102	LUKUP_2	Character	10	Relation 2 Set Into this Table
103	RLTYPE_2	Character	1	Relation Type, "P", "C" or ""
104	HOME_3	Character	10	Relation 3 Set From this Table
105	RLKEY_3	Character	40	Relational Expression 3
106	LUKUP_3	Character	10	Relation 3 Set Into this Table
107	RLTYPE_3	Character	1	Relation Type, "P", "C" or ""
108	HOME_4	Character	10	Relation 4 Set From this Table
109	RLKEY_4	Character	40	Relational Expression 4
110	LUKUP_4	Character	10	Relation 4 Set Into this Table
111	RLTYPE_4	Character	1	Relation Type, "P", "C" or ""
112	HOME_5	Character	10	Relation 5 Set From this Table
113	RLKEY_5	Character	40	Relational Expression 5
114	LUKUP_5	Character	10	Relation 5 Set Into this Table
115	RLTYPE_5	Character	1	Relation Type, "P", "C" or ""
116	HOME_6	Character	10	Relation 6 Set From this Table
117	RLKEY_6	Character	40	Relational Expression 6
118	LUKUP_6	Character	10	Relation 6 Set Into this Table
119	RLTYPE_6	Character	1	Relation Type, "P", "C" or ""
120	HOME_7	Character	10	Relation 7 Set From this Table
121	RLKEY_7	Character	40	Relational Expression 7
122	LUKUP_7	Character	10	Relation 7 Set Into this Table
123	RLTYPE_7	Character	1	Relation Type, "P", "C" or ""

ASTER - [Associacao Tecnica p Reconstruccao Nacio]

Rua Ojm, Edificio Ccm
Inhambane, Jangamo
Inhambane

Ref. No: 00141
Classification:

Phone: 023-2656
Fax:
Fiscal Year: 01/11/92

Contact
Name: Amosse Baltazar Zita
Title: Presidente
No. of Staff: 20

CARE - [Care International]

Chokwe, Chokwe
Gaza

Ref. No: 00032
Classification:

Phone:
Fax:
Fiscal Year: 01/01/00

Contact
Name:
Title:
No. of Staff: 1

Av. Martires De Mueda 596
Maputo, Maputo
Maputo City

Ref. No: 00108
Classification:

Phone: 492064/5/6
Fax: 492077
Fiscal Year: 01/01/00

Name: Peter Abrams
Title: Coordinator
No. of Staff: 1

Av. Eduardo Mondlane 768b
Nampula, Nampula
Nampula

Ref. No: 00107
Classification:

Phone: 212025
Fax: 214051
Fiscal Year: 01/07/93

Name: Dan Harvey
Title: Provincial Coordinator
No. of Staff: 22

Beira, Dondo
Sofala

Ref. No: 00042
Classification:

Phone:
Fax:
Fiscal Year: 01/01/00

Name:
Title:
No. of Staff: 1

CARITAS - [CARITAS]

Xai-Xai, Xai-Xai
Gaza

Ref. No: 00121
Classification:

Phone:
Fax:
Fiscal Year: 01/01/00

Contact
Name:
Title:
No. of Staff: 1

Maputo, Maputo
Maputo City

Ref. No: 00026
Classification:

Phone:
Fax:
Fiscal Year: 01/01/00

Name:
Title:
No. of Staff: 1

Caixa Postal 304
Nampula, Nampula
Nampula

Ref. No: 00092
Classification:

Phone: 21-3236
Fax:
Fiscal Year: 01/01/93

Name: Antonio Venancio Mucaripo
Title: Caritas Secretary
No. of Staff: 11

Beira, Dondo
Sofala

Ref. No: 00022
Classification:

Phone:
Fax:
Fiscal Year: 01/01/00

Name:
Title:
No. of Staff: 1

CASA DE GAIATO - [Casa de Gaiato]

Av 24 De Julho 1847
Maputo, Maputo
Maputo City

Ref. No: 00147
Classification:

Phone: 421407
Fax:
Fiscal Year: 01/01/00

Contact
Name: Dr Augusta
Title:
No. of Staff: 1

CCM - [Christian Council of Mozambique]

Xai-Xai, Xai-Xai
Gaza

Ref. No: 00125
Classification:

Phone:
Fax:
Fiscal Year: 01/01/00

Contact
Name:
Title:
No. of Staff: 1

Inhambane, Jangamo
Inhambane

Ref. No: 00088
Classification:

Phone:
Fax:
Fiscal Year: 01/01/00

Name:
Title:
No. of Staff: 1

Nampula, Nampula
Nampula

Ref. No: 00085
Classification:

Phone:
Fax:
Fiscal Year: 01/01/00

Name:
Title:
No. of Staff: 1

Lichinga, Lichinga
Niassa

Ref. No: 00139
Classification:

Phone:
Fax:
Fiscal Year: 01/01/00

Name:
Title:
No. of Staff: 1

ADRA - [Adventist Development & Relief Agency]

Inhambane, Jangamo
Inhambane

Ref. No: 00007
Classification:

Phone:
Fax:
Fiscal Year: 01/01/00

Contact
Name: Dwight Taylor
Title:
No. of Staff: 1

AFRICARE - [AFRICARE]

Beira, Dondo
Sofala

Ref. No: 00002
Classification:

Phone:
Fax:
Fiscal Year: 01/01/00

Contact
Name:
Title:
No. of Staff: 0

AICF - [Action Internationale Contre la Faim]

Sofala, Buzi
Sofala

Ref. No: 00129
Classification:

Phone:
Fax:
Fiscal Year: 01/01/00

Contact
Name:
Title:
No. of Staff: 1

Beira, Dondo
Sofala

Ref. No: 00003
Classification:

Phone:
Fax:
Fiscal Year: 01/01/00

Name: Hugues Burrows
Title:
No. of Staff: 1

AISPO - [Associa.Itali.perla Solidar.Tra I Populi]

Beira, Dondo
Sofala

Ref. No: 00151
Classification:

Phone:
Fax:
Fiscal Year: 01/01/00

Contact
Name: Renato Corrado
Title:
No. of Staff: 1

AMAI A PABANDA - [Amai a Pabanda]

Beira, Dondo
Sofala

Ref. No: 00158
Classification:

Phone:
Fax:
Fiscal Year: 01/01/00

Contact
Name:
Title:
No. of Staff: 1

AMRU - [Associacao Mocambicana da Mulher Rural]

Facim-Pavilhao 67 Cp 1892
Maputo, Maputo
Maputo City

Ref. No: 00142
Classification:

Phone: 427151/2-428981
Fax: 491628
Fiscal Year: 01/01/00

Contact
Name: Amelia Zambeze
Title: President
No. of Staff: 1

AMRV - [Associacao Mocambicana da Mulher Rural]

Facim Pav.75
Maputo, Maputo
Maputo City

Ref. No: 00112
Classification:

Phone: 422809
Fax: 422893
Fiscal Year: 06/08/91

Contact
Name: Amelia Zambeze
Title: National Coordinator
No. of Staff: 4

ARC - [American Red Cross]

Tete, Moatize
Tete

Ref. No: 00150
Classification:

Phone: 23315
Fax:
Fiscal Year: 01/01/00

Contact
Name: Robert Warwich
Title:
No. of Staff: 1

ARO - [Grupo Africa da Suecia]

Av. Amilcar Cabral 865 Cp 1359
Maputo, Maputo
Maputo City

Ref. No: 00078
Classification:

Phone: 421422/3
Fax: 421422
Fiscal Year: 01/01/00

Contact
Name: Anita Carlson
Title: Coordinator
No. of Staff: 1

ASEM - [Asso..Suisse Faveur L'enfante Mozambican]

Beira, Dondo
Sofala

Ref. No: 00046
Classification:

Phone:
Fax:
Fiscal Year: 01/01/00

Contact
Name:
Title:
No. of Staff: 1

United Nations Office for Humanitarian Assistance Coordination
SHAMAN: System for Humanitarian Assistance Management

WHO IS WHERE

Current to: 28 December 1993

Agency Activities
Ranked by Agency by Province by District

ACDR - [Associa. Camponese para Desenvol. Rural]

Xai-Xai, Xai-Xai
Gaza

Ref. No: 00146
Classification:

Phone:
Fax:
Fiscal Year: 01/01/00

Contact
Name: Chissano
Title: President
No. of Staff: 1

ACTION AID - [Action Aid]

Av. Herois Da Libertacao Nacional
Quelimane, Nicoadala
Zambezia

Ref. No: 00069
Classification:

Phone: 04-212134
Fax:
Fiscal Year: 01/01/93

Contact
Name: Anthony Nedley
Title: Country Director
No. of Staff: 100

ACTION NORD-SUD - [FRENCH ACTION GROUP]

Av. V.Lenin 391 Cp 4331
Maputo, Maputo
Maputo City

Ref. No: 00103
Classification:

Phone: 742676
Fax:
Fiscal Year: 01/01/00

Contact
Name: Sophie Goyet
Title: Coordinator
No. of Staff: 1

ADEMO - [Association for Deficiant Mozambicans]

Xai-Xai, Xai-Xai
Gaza

Ref. No: 00057
Classification:

Phone:
Fax:
Fiscal Year: 01/01/00

Contact
Name:
Title:
No. of Staff: 1

Nampula, Nampula
Nampula

Ref. No: 00056
Classification:

Phone:
Fax:
Fiscal Year: 01/01/00

Name:
Title:
No. of Staff: 1

Tete, Moatize
Tete

Ref. No: 00015
Classification:

Phone:
Fax:
Fiscal Year: 01/01/00

Name:
Title:
No. of Staff: 1

ADOCA - [Associacao das Donas de Casa]

Pc Alexandre Herculano 270a
Matola, Boane
Maputo

Ref. No: 00113
Classification:

Phone: 723410
Fax:
Fiscal Year: 01/01/91

Contact
Name: Giliardo/ivete
Title: Coordinator/President
No. of Staff: 8

ADPP - [Ajuda ao Povo para o Povo]

Av Mao Tse Tung 230 7th Floor
Maputo, Maputo
Maputo City

Ref. No: 00154
Classification:

Phone: 753029/742678
Fax: 490498
Fiscal Year: 01/01/00

Contact
Name: Birgit Holm
Title:
No. of Staff: 1

Province	District	Food MT	Relief & Survival MT	Relief & Survival Units	Seeds MT	Tools Units
	Mongincual	826.23	5.70	2,400	113.43	1,300
	Mossuril	57.64			12.09	
	Muecate	111.70	1.00	2,804		2,200
	Murrupula	4.30	0.06			
	Nacala Velha					500
	Namapa	546.96	10.69	3,450	57.00	3,068
	Provincial Totals	2,347.76	27.24	13,421	244.43	12,468
Niassa						
	Mandimba	75.72				
	Provincial Totals	75.72				
Sofala						
	Buzi	970.73	9.00	11,700	338.60	4,998
	Caia	22.94				860
	Chemba	303.29	0.40	16,910		3,200
	Cheringoma	3,468.36	117.94	29,746	23.60	2,292
	Chibabava	2,098.76	11.93	3,060	181.40	1,536
	Dondo	411.32				
	Gorongosa	3,397.54	14.32	28,300	194.07	4,120
	Machanga				7.50	
	Maringue	3,758.46	9.85	35,224	287.70	11,700
	Marromeu	281.66				600
	Muanza	549.91				
	Nhamatanda	1,245.89	11.30	15,740		
	Provincial Totals	16,508.86	174.74	**,***	1,032.87	29,306
Tete						
	Angonia	24.05	0.25			
	Cahora Bassa	25.12				
	Changara	186.91				
	Chifunde	58.14				
	Chiuta	273.51				
	Maravia	156.87				
	Moatize	93.38				
	Mutarara					
	Tsangano	148.30				
	Provincial Totals	967.27	0.25			
Zambezia						
	Alto Molocue	46.00		39,980		2,752
	Chinde	7.50				4,000
	Maganja Da Cost	178.02	5.02	8,843	17.05	200
	Milange	147.00				
	Mocuba	531.35	2.54	3,730		
	Mopeia	1,087.12	10.13	15,515	99.50	11,453
	Morrumbala	1,434.45	11.15	15,233	56.05	1,100
	Namacurra	60.00	4.25	5,993		
	Nicoadala				1.50	
	Pebane	141.51			29.61	1,364
	Provincial Totals	3,632.95	33.09	89,294	203.71	20,869
Country Totals		29,324.61	255.24	261,090	2,062.27	67,112

Note: R&S (Relief and Survival) items - Soap is reported in MT, all other items in Units.

United Nations Office for Humanitarian Assistance Coordination

SHAMAN: System for Humanitarian Assistance Management

RELIEF DISPATCHES TO RENAMO AREAS

Compiled: 29 December 1993

Summary Totals by Province and District
Ranked by Province by District by Category

REPORT SUMMARY TOTALS			
Total Food MT:	29,324.61		
Total Other MT:	2,540.62		
Total MT:	31,865.23	Total Units:	364,870

UNOHAC — RELIEF DISPATCHES TO RENAMO AREAS - From October 1992 to 29 December 1993 — Maputo

Province	District	Food MT	Relief & Survival MT	Relief & Survival Units	Seeds MT	Tools Units
Gaza						
	Bilene	8.15				
	Chibuto	1,088.35	8.23	10,257	99.00	80
	Chicualacuala	2.65				
	Chigubo	74.01	0.29	320	0.25	
	Mabalane	14.87		30		
	Provincial Totals	1,188.03	8.52	10,607	99.25	80
Inhambane						
	Homoine	88.50		5,000		
	Panda	100.00				
	Vilanculos	67.92				
	Provincial Totals	256.42		5,000		
Manica						
	Barue	83.25				
	Gondola	370.17				
	Macossa	722.54			25.00	
	Manica	22.05				
	Mossurize	53.55				
	Sussundenga	2,051.97	8.00		393.58	2,250
	Tambara	568.34				
	Provincial Totals	3,871.87	8.00		418.58	2,250
Maputo						
	Magude	102.57	0.90	560	52.18	1,160
	Manhica	30.80		32	0.65	32
	Marracuene	57.48	0.17	97	1.95	97
	Matutuine	250.23	2.33	1,406	8.59	850
	Moamba	33.10				
	Namaacha	1.55		3		
	Provincial Totals	475.73	3.40	2,098	63.37	2,139
Nampula						
	Angoche	63.70	[illegible]	957	15.00	

ency	Province	District	Category	Quantity MT	Quantity Units			
		Buzi	Food	112.41				
		Cheringoma	R&S		11			
		Marromeu	Food	280.21				
		Marromeu	Seeds		1,817			
Provincial Totals		Total Food MT:	392.62	Total Other MT:	0.00	Total MT:	392.62	Total Units: 1,42[illegible]
Agency Totals		Total Food MT:	392.62	Total Other MT:	0.00	Total MT:	392.62	Total Units: 1,42[illegible]
[illegible]A								
	Nampula							
		Angoche	Food	48.[illegible]				
		Angoche	Seeds	15.[illegible]				
		Malema	Food	[illegible]				
		Mongincual	Food	10.[illegible]				
		Mongincual	Seeds	3.[illegible]				
		Mossuril	Food	57.64				
		Mossuril	Seeds	12.09				
		Muecate	Food	111.[illegible]				
Provincial Totals		Total Food MT:	234.66	Total Other MT:	30.59	Total MT:	265.25	Total Units: [illegible]
Agency Totals		Total Food MT:	234.66	Total Other MT:	30.59	Total MT:	265.25	Total Units: [illegible]
[illegible]RC								
	Gaza							
		Chibuto	Food	1,081.43				
		Chibuto	R&S	8.23	7,961			
		Chibuto	Seeds	99.00				
		Chigubo	Food	45.74				
		Chigubo	R&S	0.29	320			
		Chigubo	Seeds	0.25				
Provincial Totals		Total Food MT:	1,127.17	Total Other MT:	107.77	Total MT:	1,234.94	Total Units: 8,281
	Manica							
		Macossa	Food	240.00				
		Sussundeng	Food	1,409.20				
		Sussundeng	R&S	8.00				
		Sussundeng	Seeds	265.20				
		Sussundeng	Tools		2,250			
		Tambara	Food	61.00				
Provincial Totals		Total Food MT:	1,710.20	Total Other MT:	273.20	Total MT:	1,983.40	Total Units: 2,25[illegible]
	Maputo							
		Magude	Food	46.76				
		Magude	R&S	0.90	550			
		Magude	Seeds	6.57				
		Magude	Tools		650			
		Matutuine	Food	224.12				
		Matutuine	R&S	2.33	1,000			
		Matutuine	Seeds	8.59				
		Matutuine	Tools		850			
		Moamba	Food	21.92				
Provincial Totals		Total Food MT:	292.70	Total Other MT:	18.39	Total MT:	311.09	Total Units: 3,05[illegible]
	Nampula							
		Angoche	Food	15.40				
		Angoche	R&S	0.40	957			
		Mogovolas	Food	730.23				
		Mogovolas	R&S	9.39	3,600			
		Mogovolas	Seeds	47.00				
		Mogovolas	Tools		2,600			
		Mongincual	Food	712.31				
		Mongincual	R&S	5.70	2,900			
		Mongincual	Seeds	29.90				
		Mongincual	Tools		1,300			
		Muecate	R&S	1.00	4			
		Murrupula	Food	4.30				
		Murrupula	R&S	0.06				
		Namapa	Food	546.96				
		Namapa	R&S	10.69	3,450			
		Namapa	Seeds	57.00				
		Namapa	Tools		3,068			
Provincial Totals		Total Food MT:	2,009.20	Total Other MT:	161.14	Total MT:	2,170.34	Total Units: 17,[illegible]
	Sofala							
		Buzi	Food	854.[illegible]				
		Buzi	R&S	[illegible]	11,700			
		Buzi	Seeds	323.[illegible]				
		Buzi	Tools		4,998			
		Chemba	Food	93.[illegible]				
		Chemba	R&S	[illegible]	16,310			
		Cheringoma	Food	186.[illegible]				

gency	Province	District	Category	Quantity MT	Quantity Units			
Provincial Totals		Total Food MT:	13.26	Total Other MT:	0.00	Total MT:	13.26	Total Units:
	Sofala							
		Buzi	Seeds	15.00				
		Caiabava	Seeds	15.00				
		Goringosa	Seeds	7.50				
		Maringue	Seeds	7.50				
Provincial Totals		Total Food MT:	0.00	Total Other MT:	45.00	Total MT:	45.00	Total Units:
Agency Totals		Total Food MT:	136.29	Total Other MT:	45.00	Total MT:	181.29	Total Units:
CH								
	Inhambane							
		Homoine	Food	88.50				
Provincial Totals		Total Food MT:	88.50	Total Other MT:	0.00	Total MT:	88.50	Total Units:
Agency Totals		Total Food MT:	88.50	Total Other MT:	0.00	Total MT:	88.50	Total Units:
ONCERN								
	Nampula							
		Muecate	R&S		2,500			
Provincial Totals		Total Food MT:	0.00	Total Other MT:	0.00	Total MT:	0.00	Total Units: 2,5
Agency Totals		Total Food MT:	0.00	Total Other MT:	0.00	Total MT:	0.00	Total Units: 2,5
HA								
	Zambezia							
		Morrumbala	R&S	5.00	6,773			
Provincial Totals		Total Food MT:	0.00	Total Other MT:	5.00	Total MT:	5.00	Total Units: 6,7
Agency Totals		Total Food MT:	0.00	Total Other MT:	5.00	Total MT:	5.00	Total Units: 6,7
HA/ITALIAN COOPERAT								
	Sofala							
		Maringue	R&S		1,100			
Provincial Totals		Total Food MT:	0.00	Total Other MT:	0.00	Total MT:	0.00	Total Units: 1,1
Agency Totals		Tot.. Food MT:	0.00	Total Other MT:	0.00	Total MT:	0.00	Total Units: 1,1
HA/UNDRO								
	Sofala							
		Cheringoma	R&S	12.56	22,434			
		Maringue	R&S	6.00	9,724			
		Maringue	Tools		100			
Provincial Totals		Total Food MT:	0.00	Total Other MT:	18.56	Total MT:	18.56	Total Units: 32,2
	Zambezia							
		Maganja Da	R&S	5.00	8,840			
		Mocuba	R&S	2.54	3,730			
		Mopeia	R&S	10.03	15,515			
		Morrumbala	R&S	6.00	8,460			
		Namacurra	R&S	4.25	5,993			
Provincial Totals		Total Food MT:	0.00	Total Other MT:	27.82	Total MT:	27.82	Total Units: 42,5
Agency Totals		Total Food MT:	0.00	Total Other MT:	46.38	Total MT:	46.38	Total Units: 74,7
CCN								
	Tete							
		Angonia	Food	24.05				
		Angonia	R&S	0.25				
		Chiuta	Food	3.11				
		Tsangano	Food	140.80				
Provincial Totals		Total Food MT:	167.96	Total Other MT:	0.25	Total MT:	168.21	Total Units:
Agency Totals		Total Food MT:	167.96	Total Other MT:	0.25	Total MT:	168.21	Total Units:
OD FOR THE HUNGRY								
	Sofala							

United Nations Office for Humanitarian Assistance Coordination
SHAMAN: System for Humanitarian Assistance Management

RELIEF DISPATCHES TO RENAMO AREAS

Current to: 29 December 1993

Ranked by Agency by Province by District by Category

REPORT SUMMARY TOTALS			
Total Food MT:	29,324.61		
Total Other MT:	2,547.82		
Total MT:	31,872.43	Total Units:	364,870.20

Note: R&S (Relief and Survival) items - Soap is reported in MT, all other items in units

OHAC — RELIEF DISPATCHES TO RENAMO AREAS - From October 1992 to 29 December 1993 — Maputo

gency	Province	District	Category	Quantity MT	Quantity Units			
CTION AID								
	Zambezia							
		Maganja Da	Food	61.20				
		Maganja Da	Seeds	17.05				
		Maganja Da	Tools		200			
		Pebane	Food	141.51				
		Pebane	Seeds	29.61				
		Pebane	Tools		1,364			
Provincial Totals		Total Food MT:	202.71	Total Other MT:	46.66	Total MT:	249.37	Total Units: 1,564
Agency Totals		Total Food MT:	202.71	Total Other MT:	46.66	Total MT:	249.37	Total Units: 1,564
ORA								
	Inhambane							
		Vilanculos	Food	67.92				
Provincial Totals		Total Food MT:	67.92	Total Other MT:	0.00	Total MT:	67.92	Total Units: 0
Agency Totals		Total Food MT:	67.92	Total Other MT:	0.00	Total MT:	67.92	Total Units: 0
APUCHINO MISSIONARY								
	Zambezia							
		Chinde	Food	7.50				
		Mopeia	Food	24.92				
		Mopeia	R&S	0.10				
		Mopeia	Seeds	9.70				
		Morrumbala	Food	41.10				
		Morrumbala	R&S	0.15				
		Morrumbala	Seeds	56.05				
		Nicoadala	Seeds	1.50				
Provincial Totals		Total Food MT:	73.52	Total Other MT:	67.50	Total MT:	141.02	Total Units: 0
Agency Totals		Total Food MT:	73.52	Total Other MT:	67.50	Total MT:	141.02	Total Units: 0
RITAS								
	Gaza							
		Bilene	Food	8.15				
		Mabalane	Food	14.87				
		Mabalane	R&S		30			
Provincial Totals		Total Food MT:	23.02	Total Other MT:	0.00	Total MT:	23.02	Total Units: 30
	Inhambane							
		Panda	Food	100.00				
Provincial Totals		Total Food MT:	100.00	Total Other MT:	0.00	Total MT:	100.00	Total Units:
	Maputo							
		Matutuine	Food	4.71				
		Namaacha	Food	4.85				

OHAC SHAMAN — POPULATION - 1980, Dec 1992, Dec 1995 — MOZAMBIQUE

Province	District	1980	December 1992	December 1995	1992 No of Beneficiaries
Sofala					
	Maringue		66,854	82,035	107,000
	Marromeu		81,515	97,096	50,000
	Muanza		28,398	33,027	
	Nhamatanda		80,705	92,458	139,000
	Sofala Totals		1,335,839	1,545,063	854,500
Tete					
	Angonia		65,905	114,110	6,300
	Cahora Bassa		58,641	63,912	39,700
	Changara		92,456	110,533	72,000
	Chifunde		20,199	29,288	2,700
	Chiuta		46,948	53,682	17,300
	Macanga		22,189	34,196	4,400
	Magoe		20,216	24,585	28,600
	Maravia		51,258	58,138	12,000
	Moatize		233,579	295,311	32,000
	Mutarara		59,905	84,442	15,000
	Tsangano		45,889	62,214	900
	Zumbu		30,974	39,715	15,000
	Tete Totals		748,159	970,126	243,900
Zambezia					
	Alto Molocue		99,327	104,504	20,000
	Chinde		189,142	193,498	65,000
	Gile		124,055	137,973	20,000
	Gurue		255,773	299,121	6,000
	Ile		313,741	317,522	60,000
	Inhassunge		88,209	91,639	20,000
	Lugela		144,365	146,392	55,000
	Maganja Da Costa		223,460	229,291	67,300
	Milange		202,228	232,591	61,500
	Mocuba		221,291	250,726	25,000
	Mopeia		132,089	136,287	79,000
	Morrumbala		262,886	301,450	99,000
	Namacurra		132,219	134,989	50,000
	Namarroi		79,212	84,255	10,000
	Nicoadala		342,374	391,638	35,000
	Pebane		174,667	177,565	52,000
	Zambezia Totals		2,985,038	3,229,441	724,500
	Country Totals		15,594,447	17,669,781	3,659,100

Printed: 30/12/93 - 08:51

OHAC SHAMAN

POPULATION - 1980, Dec 1992, Dec 1995

MOZAMBIQUE

Province	District	1980	December 1992	December 1995	1992 No of Beneficiaries
Manica					
	Manica		123,980	160,890	83,100
	Mossurize		53,643	68,065	23,000
	Sussundenga		[illegible]	59,033	60,500
	Tambara		35,208	41,092	19,500
	Manica Totals		659,359	821,505	373,800
Maputo					
	Boane		473,746	541,467	34,200
	Magude		80,716	118,639	52,300
	Manhica		151,720	168,594	88,600
	Marracuene		56,514	66,447	31,500
	Matutuine		74,918	84,691	21,700
	Moamba		493,582	544,396	28,000
	Namaacha		24,502	34,787	22,700
	Maputo Totals		1,355,698	1,559,021	279,000
Maputo City					
	Maputo		970,650	1,186,377	
	Maputo City Totals		970,650	1,186,377	
Nampula					
	Angoche		249,032	281,263	24,000
	Erati		77,653	78,267	6,100
	Lalaua		36,906	39,621	20,500
	Malema		99,227	103,164	7,900
	Meconta		93,907	98,218	20,300
	Mecuburi		102,286	116,727	26,500
	Memba		194,838	202,225	27,300
	Mogovolas		246,368	254,716	13,400
	Moma		206,889	210,444	
	Monapo		222,134	225,822	10,300
	Mongincual		100,493	105,420	43,000
	Mossuril		127,654	158,159	17,300
	Muecate		52,695	55,379	18,800
	Murrupula		105,746	111,246	10,800
	Nacala Velha		285,877	321,941	8,000
	Namapa		244,507	245,567	14,500
	Nampula		409,567	450,423	8,500
	Ribaue		101,352	104,761	4,000
	Nampula Totals		2,957,131	3,163,363	281,200
Niassa					
	Cuamba		83,961	97,001	13,900
	Lago		46,163	47,508	3,100
	Lichinga		149,908	170,001	1,400
	Majune		22,520	27,690	8,400
	Mandimba		57,961	78,520	9,000
	Marrupa		34,250	36,129	4,800
	Maua		53,083	59,452	17,200
	Mavago		17,349	19,711	5,500
	Mecanhelas		90,197	93,249	13,000
	Mecula		13,336	18,627	2,600
	Metarica		26,692	29,301	2,500
	Muembe		19,975	23,449	400
	N'gauma		34,928	39,810	3,500
	Nipepe		38,485	43,623	11,500
	Sanga		28,678	30,598	4,500
	Niassa Totals		717,486	814,669	101,300
Sofala					
	Buzi		200,403	212,063	92,800
	Caia		74,412	99,585	51,200
	Chemba		42,238	63,422	22,300
	Cheringoma		43,512	74,370	21,100
	Chibabava		85,866	90,762	33,100
	Dondo		474,307	539,618	159,100
	Gorongosa		71,667	91,604	75,200
	Machanga		65,962	68,995	43,200

228

United Nations Office for Humanitarian Assistance Coordination
SHAMAN: System for Humanitarian Assistance Management

POPULATION - 1980, DEC '92, DEC '95

Current to: 30 December 1993

1980 Census Data, December 1992 GCM, December 1995 GCM Projection
Ranked by Province by District

OHAC SHAMAN — POPULATION - 1980, Dec 1992, Dec 1995 — MOZAMBIQUE

Province	District	1980	December 1992	December 1995	1992 No of Beneficiaries
	Muindumbe		55,949	56,588	3,800
	Totals		55,949	56,588	3,800
Cabo Delgado					
	Ancuabe		81,614	87,556	5,000
	Balama		108,378	119,127	15,000
	Chiure		160,245	167,842	8,500
	Macomia		77,238	78,623	2,300
	Mecufi		38,908	41,448	8,500
	Meluco		30,680	35,516	10,000
	Mocimboa Da Praia		68,676	71,971	
	Montepuez		160,741	191,798	17,500
	Mueda		106,339	113,028	3,000
	Namuno		149,665	170,166	29,000
	Nangade		43,236	47,546	2,200
	Palma		34,202	38,460	
	Pemba		104,705	121,260	3,500
	Quissanga		40,195	43,679	18,700
	Cabo Delgado Totals		1,204,822	1,328,020	123,200
Gaza					
	Bilene		135,851	138,654	35,500
	Chibuto		241,874	286,740	52,000
	Chicualacuala		38,472	47,639	25,500
	Chigubo		40,891	49,963	36,000
	Chokwe		182,940	226,881	75,000
	Guija		87,535	95,047	30,000
	Mabalane		36,605	45,898	26,000
	Mandlakazi		142,876	150,320	47,600
	Massangena		19,669	31,003	21,000
	Massingir		37,290	43,447	21,500
	Xai-Xai		204,041	234,852	30,000
	Gaza Totals		1,168,044	1,350,444	400,100
Inhambane					
	Funhalouro		53,491	68,923	20,000
	Govuro		72,742	89,325	17,500
	Homoine		229,684	278,889	13,900
	Inharrime		82,835	91,086	20,000
	Inhassoro		97,872	111,450	26,000
	Jangamo		247,684	279,628	11,500
	Mabote		33,400	54,589	30,000
	Massinga		216,020	219,815	21,400
	Morrumbene		117,801	133,072	13,300
	Panda		70,633	81,449	40,200
	Vilanculos		116,717	129,135	35,000
	Zavala		97,393	107,803	25,000
	Inhambane Totals		1,436,272	1,645,164	273,800
Manica					
	Barue		52,128	69,380	46,000
	Gondola		210,755	257,573	55,000
	Guro		55,302	72,054	46,800
	Machaze		58,059	66,877	36,900
	Macossa		22,365	26,541	8,000

Note: [illegible]

NAMPULA

28

District	Location	Date	Reported by: UNOHAC REP NAMPULA
MECONTA	7 DE ABRIL	17/05/93	Map Sheet: 44 Grid Reference: Latitude: Longitude: ° 39.7' E

Remarks: Road between 7 de Abril and Muessia is I believed to be mined.

29

District	Location	Date	Reported by: UNOHAC REP NAMPULA
MECONTA	CORRANE	17/05/93	Map Sheet: 44 Grid Reference: Latitude: Longitude: ° 39.6' E

Remarks: Mined areas are within the residential I area of Corrane and close to the I maternity section of the hospital.

31

District	Location	Date	Reported by: UNOHAC REP NAMPULA
MECONTA	CORRANE	17/05/93	Map Sheet: 44 Grid Reference: Latitude: Longitude: ° 39.6' E

Remarks: The road between Mecua and Corrane is I believed to be mined.

12

District	Location	Date	Reported by: DAVID HEWITSON
MECONTA	MECUA	20/05/93	Map Sheet: Grid Reference: Latitude: Longitude: ° 39.3' E

Source: Local inhabitants

Reported Mines: Close to the road heading south through the village is a group of concrete buildings, three or four on either side of the road, approximately five metres from the road. It is reported that there are mines close to the buildings on the sides away from the road.

Suspected Mines: Local people are now clearing the buildings ready for use. They are confident that there are no mines between the buildings and the road. There may well be more casualties. These buildings should not be approached.

Remarks: In approximately 1985 FRELIMO forces left the town, which was then occupied by RENAMO. When they in turn left, they laid mines arond the buildings to prevent their re-use. At least one casualty has been suffered close to these buildings (January, 1993).

[see sketch map]

Even though some ground has been cleared, without casualty so far, there is no guarantee that mining only took place on the far side of the buildings, away from the road. The supposed reason for the mining was denial of the use of the buildings. It is therefore likely that mines were placed between the buildings and the road.

30

District	Location	Date	Reported by: UNOHAC REP NAMPULA
MECONTA	MECUA	17/05/93	Map Sheet: 44 Grid Reference: Latitude: Longitude: ° 39.4' E

Remarks: The road between Mecua and Corrane is I believed to be mined.

Note: the section of road is defined by I the positions in report numbers 30 and I 31.

27

District	Location	Date	Reported by: UNOHAC REP NAMPULA
ILHA DE MOZAMBIQUE	MUTOMONHO	14/05/93	Map Sheet: 45 Grid Reference: Latitude: Longitude: ° 40.5' E

Remarks: Suspected zone close to the entrance to I the Lunga area.

MANICA

47

District	Location	Date	Reported by: ENRIQUE PORTILLO
MANICA	MAVONDE	17/08/93	Map Sheet: Grid Reference: Latitude: Longitude:

Source: RENAMO.

Reported Mines: Small mine field identified and marked by I RENAMO.

Remarks: There is a small mine field around the I middle of Mavonde.

NAMPULA

19

District	Location	Date	Reported by: AAA
ANGOCHE		24/05/93	Map Sheet: Grid Reference: Latitude: Longitude:

Source: Local NGO; AAA

Reported Mines: The southern bank of the river Metomode running east from Namaponda and the river running east from Mutucute.

16

District	Location	Date	Reported by: AAA
MALEMA		24/05/93	Map Sheet: Grid Reference: Latitude: Longitude: ° 37.4' E

Source: Local NGO; AAA

Reported Mines: River bank where the road north from Macalia meets the river Lurio reported mined.

Suspected Mines: See Remarks.

Remarks: The river Lurio formed a barrier between Colonial and FRELIMO forces during the war of independence. All areas around the western borders of Nampula province should be suspect.

17

District	Location	Date	Reported by: AAA
MALEMA	MALEMA	24/05/93	Map Sheet: Grid Reference: Latitude: Longitude:

Source: Local NGO; AAA

Reported Mines: The road between Malema and Nacathe, 2/3 of the way to Mutuali, is reported to be mined.

Remarks: This is the main road leading west from Nampula towards Niassa and Malawi beyond. It is believed that the first 10km west of Malema are safe. There are also bridges down along this road.

United Nations Office for Humanitarian Assistance Coordination

SHAMAN: System for Humanitarian Assistance Management

MINE ASSESSMENT & SURVEY

Current to: 23 December 1993

Ranked by Province by District

REPORT SUMMARY TOTALS	
Total Mines Found:	16
Total Mines Cleared:	4

	Activity Total:	300,000
6.0.06	Support to NAR	Needs
	Financial assistance to NAR	4,000,000
	Activity Total:	4,000,000
6.0.07	Ministry of Education Emergency Response	Needs
	Provision for Technical Assistance	300,000
	Training Seminars upgrading 60 teachers	18,000
	Transport and Logistics	295,000
	Activity Total:	613,000
6.0.08	Support to Social Welfare and CIDC	Needs
	Advocacy and social communication	300,000
	Institutional support	300,000
	Local recruitment of central task force & costs	300,000
	Re-integration of vulnerable groups and CIDC	1,220,000
	Rehabilitation of social action facilities	1,200,000
	Support to NGO network	900,000
	Technical assistance	130,000
	Activity Total:	4,350,000
Sector Total:		18,023,000
ategory Total:		18,023,000

NEEDS SUMMARY

Grand Total Needs:609,688,760

Printed: 29/12/93 - 12:2

4.7 Mine Clearance		
4.7.01	Mine Survey & Road Mine Clearance	Needs
	Cost of survey contract	420,000
	Mine clearance contracts	25,000,000
	Monitoring and quality assurance	1,400,000
	Salary and equipment costs	442,000
	Activity Total:	27,302,000
4.7.02	Mine Awareness & Clearance Training	Needs
	Building and miscelaneous costs	655,640
	Equipment costs	253,695
	Personnel Costs	2,206,000
	UNHCR mine awareness campaign	500,000
	Activity Total:	4,315,535
Sector Total:		31,617,535
ategory Total:		327,002,099

BALANCE OF PAYMENTS AND BUDGET SUPPORT		
5.0 Balance of Payment Support		
5.0.01	Market Food Aid	Needs
	Maize	22,000,000
	Rice	18,956,000
	Wheat	18,102,000
	Activity Total:	59,058,000
Sector Total:		59,058,000
Category Total:		59,058,000

INSTITUTIONAL SUPPORT		
6.0 Institutional Support		
6.0.01	Food Security & Early Warning Systems	Needs
	Improvement of communication network	95,000
	Provincial food security data collection network	70,000
	Rehabilitation of meteorological stations	500,000
	Activity Total:	665,000
6.0.02	Support to CPEs and CDEs	Needs
	Support to CENE/CPEs and CDEs	2,000,000
	Activity Total:	2,000,000
6.0.03	Institutional Support to DPCCN	Needs
	Administration and finance	800,000
	Planning and information	700,000
	Support to provincial DPCCN	600,000
	Technical assistance to LSU	3,000,000
	Activity Total:	5,100,000
6.0.04	MCH Emergency Response	Needs
	Central task force, duty travel and local costs	625,000
	Equipment, transport, communications and offices	30,000
	General expenses, office costs, documentation	100,000
	International recruitment	170,000
	Local contracts	30,000
	Short term consultancies	40,000
	Activity Total:	995,000
6.0.05	National NGOs	Needs

	Well and borehole const.equipment and costs	3,[illegible]46,000
	Activity Total:	14,425,000
Sector Total:		14,425,000

4.4 Roads and Bridges

4.4.01	Roads and Bridges	Needs
	Construction - DNEP and NGOs	20,[illegible]00,000
	Survey, design and supervision	14,[illegible]00,000
	Activity Total:	34,[illegible]00,000
Sector Total:		34,[illegible]00,000

4.5 Education

4.5.01	Education and Teachers Training	Needs
	Functional rehabilitation of 1,000 classrooms	5,000,000
	Provision of 500 teachers kits	75,000
	Provision of material and equipment kits	3,[illegible]00,000
	Purchase school text books	1,[illegible]00,000
	Activity Total:	9,575,000
Sector Total:		9,575,000

4.6 Multi-sectoral Programs

4.6.01	Area Based Programmes at Community Level	Needs
	Area-based programmes at community level	25,000,000
	Activity Total:	25,000,000

4.6.02	Support to Returnee Areas (QIPS)	Needs
	QUIPS	13,987,000
	Activity Total:	13,987,000

4.6.03	Institutional Support for Demob Soldiers	Needs
	CORE's Programme Development	380,000
	Information/Counselling Services	1,600,000
	Veterans Association(s)	500,000
	Vocational Training Instit.	2,583,000
	Activity Total:	5,063,000

4.6.04	Training for Demobilized Soldiers	Needs
	Vocational/Technical/Entrepreneurial/Training	6,484,000
	Activity Total:	6,484,000

4.6.05	Kits and Credit Scheme for Demobilized S	Needs
	Follow-up activity loans	3,500,000
	Group/association loans	2,000,000
	Kits and Credit Admin Costs	500,000
	Small-business loans	4,000,000
	Vocational/Activity kits	10,000,000
	Activity Total:	20,000,000

4.6.06	Employment for Demobilized Soldiers	Needs
	Job identification and Support	1,400,000
	Labour-Intensive Work District Level Fund	1,000,000
	Vegetable Kits	765,250
	Activity Total:	3,[illegible]65,250
Sector Total:		73,[illegible]49,250

	Recruitment and transportation of extension agents	1,000,000
	Training and logistics	150,000
	Activity Total:	1,500,000

4.1.08	Reserve Grain Silo, Beira	Needs
	Contractual services	1,098,000
	Operational expenses	45,000
	Activity Total:	1,143,000

Sector Total:		111,145,314

4.2 Health

4.2.01	Stockpiles of Drugs Reagents	Needs
	Drugs for response to epidemics	400,000
	Drugs for rural hospitals, in-patient level	2,000,000
	EDF and WHO emergency for PHC level	1,600,000
	Support to MEDIMOC's imports	5,000,000
	Activity Total:	9,000,000

4.2.02	Cold Chain Equipment/Vaccines	Needs
	Cold-chain equipment and other supplies	400,000
	Vaccines and consumables	400,000
	Activity Total:	1,400,000

4.2.03	Support to Provinc. Health Directorates	Needs
	Equipment: transport, communication, office	400,000
	General operating expenses	500,000
	Local costs	800,000
	Technical assistance	1,000,000
	Training and travel expenses	700,000
	Activity Total:	3,500,000

4.2.04	Rehabilitation of Health Network	Needs
	General operating expenses	1,200,000
	Hospital equipment and furniture	750,000
	Medical supplies and sterilization equipment	3,250,000
	Rehabilitation of health facilities	6,100,000
	Technical assistance and NGO contracts	2,000,000
	Activity Total:	13,300,000

4.2.05	Rural Health Care	Needs
	Contract of NGOs Services: Salaries, commod.,etc.	18,000,000
	Support to provincial health authorities	500,000
	Activity Total:	18,500,000

4.2.06	Prosthetics Assistance	Needs
	Material and equipment for workshops	1,300,000
	NGOs contracts and technical assistance	2,500,000
	Operating expenses, duty travel for supervision	800,000
	Rehabilitation works	800,000
	Training activities	350,000
	Vehicles, communications, office equip., logistics	250,000
	Activity Total:	6,000,000

Sector Total:		51,900,000

4.3 Water

4.3.01	Water Supply for Resettlement	Needs
	AFRIDEV handpumps	1,224,000
	Percussion drilling rigs & mand drilling rigs	1,950,000
	Sanitation	1,250,000
	Vehicles and spares (large/small trucks - 70)	7,125,000

3.3 Transport of People	
3.3.01 Transport of Vulnerable Groups	Needs
Office costs	1,124,000
Operational costs	5,225,000
Staff	1,145,469
Activity Total:	7,394,469
Sector Total:	7,394,469

3.4 Non-Food Relief	
3.4.01 Returnees and Internally Displaced	Needs
Logistics and storage	1,146,750
Purchase of relief and survival items	4,587,000
Purchase of water jars, tents and plastic sheeting	2,000,000
Activity Total:	7,733,750
3.4.02 Emergency Stockpile Project	Needs
Logistics and distribution	2,977,500
New equipped warehouses	595,000
Purchase of relief items	8,310,000
Technical assistance and operation costs	826,800
Activity Total:	12,711,300
Sector Total:	19,545,050
Category Total:	134,902,187

4 RESTORATION OF ESSENTIAL SERVICES

4.1 Agriculture

4.1.02 Seed for Family Sector	Needs
Logistic and administrative support	500,000
Promotion of local seed multiplication	1,000,000
Purchase of seeds	37,114,000
Seed quality control	100,000
Transport and storage	3,925,000
Activity Total:	42,639,000

4.1.03 Supply of Agricultural Tools	Needs
Purchase of tools	47,831,314
Transport and storage	2,000,000
Activity Total:	49,831,314

4.1.04 Pest Control and Seed Inspection	Needs
Pest control	1,100,000
Seed inspection	250,000
Activity Total:	1,350,000

4.1.05 Cassava Production	Needs
Support to INIA's and district centers	1,100,000
Transportation and supervision	400,000
Activity Total:	1,500,000

4.1.06 Marketing of Agricultural Produce	Needs
Local purchase (including ITSH	12,772,000
Operation funds (at average USD 20,000/province	200,000
Training and orientation	50,000
Activity Total:	13,022,000

4.1.07 Cassa Agricultural Extension Programme	Needs
Local [illegible]	[illegible]

	Activity Total:	1,312,747

2.0.10	Information & Social Reintegrat.	Needs
	Literacy & Printed Materials	143,500
	Personnel	94,800
	Radio	107,000
	Theatre, Cultural & Sports Activities	142,500
	Activity Total:	477,800

2.0.11	Demob & Reintegrat. of Vulnerable Groups	Needs
	Family Reunification	960,000
	Logistics and Monitoring	97,000
	Personnel	174,000
	Registration - Information for pension disabled	119,000
	Support for Reintegration	275,000
	Activity Total:	1,625,000

Sector Total:		59,703,474
Category Total:		59,703,474

EMERGENCY RELIEF

3.1 Relief Food Aid

3.1.01	Food Relief Distributions	Needs
	Provision of cereals	29,704,700
	Provision of non-cereals	21,123,200
	Activity Total:	50,827,900

3.1.02	Food for Work	Needs
	Food for work programmes	
	Activity Total:	0

3.1.03	Nutritional Rehabilitation	Needs
	Commodities for hospital diet	2,500,000
	Commodities for therapeutic feeding	7,190,000
	Activity Total:	9,690,000

Sector Total:		60,517,900

3.2 Logistics

3.2.01	Road, Rail and Sea Transport	Needs
	Road, rail and sea transport	41,576,769
	Activity Total:	41,576,769

3.2.02	Airlift Operations	Needs
	Airlift operations	2,500,000
	Activity Total:	2,500,000

3.2.03	Maintenance and Storage	Needs
	Fuel and lubricants	768,000
	Spares	600,000
	Storage	2,000,000
	Activity Total:	3,368,000

Sector Total:		47,444,769

: REPATRIATION OF REFUGEES

1.0 Repatriation Operation

1.0.01	Repatriation Operation	Needs
	Logistical Support to Repatriation	11,000,000
	Activity Total:	11,000,000
Sector Total:		11,000,000
Category Total:		11,000,000

DEMOBILIZATION

2.0 Demobilization

2.0.01	Supply of Food For Demobilization	Needs
	Food	4,714,9[illegible]
	Milling	290,000
	Ocean Freght and ITSH	2,851,554
	Transport & Storage	60,294
	Activity Total:	7,916,805

2.0.02	Supply of Non-food Items for Demobilizat	Needs
	Supply of Non-Food Items & Accomodation	3,161,500
	Activity Total:	3,161,500

2.0.03	Health Care for Demobilization	Needs
	Equipment	500,000
	General Operational Expenses	300,000
	Medical Spplies	1,000,000
	Miscellaneous	476,022
	Salaries	1,700,000
	Activity Total:	3,976,022

2.0.04	Water Supply and Sanitation for Demobil	Needs
	Implementation of works	282,637
	Purchase supplies	219,829
	Technical assistance	251,234
	Activity Total:	753,700

2.0.05	Civilian Clothing for Demobilization	Needs
	Civilian Clothing	2,513,430
	Transport	208,257
	Activity Total:	2,721,687

2.0.06	Registration of Demobilizing Soldiers	Needs
	Registration & Database	300,000
	Activity Total:	300,000

2.0.07	Home Transportation for Demobilized	Needs
	General administration	300,000
	Office and equipment	1,016,000
	Operational costs	11,962,210
	Personnel costs	1,810,000
	Activity Total:	15,088,210

2.0.08	Demobilization Subsidies	Needs
	Demobilization subsidies	22,470,000
	Activity Total:	22,470,000

2.0.09	Technical Unit for Demobilization	Needs
	Technical Unit	1,110,[illegible]

United Nations Office for Humanitarian Assistance Coordination
SHAMAN: System for Humanitarian Assistance Management

CONSOLIDATED HUMANITARIAN PROGRAMME NEEDS SUMMARY

Current to: 29 December 1993

Ranked by Activity Number

REPORT SUMMARY TOTALS
NEEDS SUMMARY
Grand Total Needs:

ADRA

Contact: Taylor
Donor Project No:

Activity Number: 3.1.01 Reference No: 0027
Category: EMERGENCY RELIEF
Sector: Relief Food Aid
Title: Food Relief Distributions
Starting From: 1 October 1992 To: 30 September 1993
Implementing Agency: ADRA

Carry Overs 92/93:	
Committments 93/94:	114,000
Total:	114,000

Beneficiaries Number: 76,000
Classification: Displaced & Returnees
Location: Inhambane

Disbursements or Obligated to Disburse
(Usually means a signed contract)
(Estimate if necessary)
Amount: 114,000

Note *To update, just write on a photocopy and send the changes to us. Please, fill in "Obligated to Disburse", "Location" details, etc where missing. Thank you.*

Remarks: ADRA's fiscal year is 1 Oct to 30 Sep.

(Include any lines which will help the reader to understand the commitment information)

ADRA

Contact: Taylor
Donor Project No:

Activity Number: 3.1.02 Reference No: 0031
Category: EMERGENCY RELIEF
Sector: Relief Food Aid
Title: Food for Work
Starting From: 1 October 1992 To: 30 September 1993
Implementing Agency: ADRA

Carry Overs 92/93:	
Committments 93/94:	30,000
Total:	30,000

Beneficiaries Number: 2,500
Classification: General Population
Location: Inh

Disbursements or Obligated to Disburse
(Usually means a signed contract)
(Estimate if necessary)
Amount: 22,500

Note: *To update, just write on a photocopy and send the changes to us. Please, fill in "Obligated to Disburse", "Location" details, etc where missing. Thank you.*

Remarks:

(Include any lines which will help the reader to understand the commitment information)

United Nations Office for Humanitarian Assistance Coordination

SHAMAN: System for Humanitarian Assistance Management

CONTRIBUTIONS BY DONOR

Current to: 28 December 1993

Please fill in missing information and/or send corrections. Thank you.

ACTION AID

Activity Number: 3.1.01 Reference No: 0021
Category: EMERGENCY RELIEF
Sector: Relief Food Aid
Title: Food Relief Distributions
Starting From: 1 September 1992 To: 30 June 1993
Implementing Agency: ACTION AID

Beneficiaries Number:
Classification: Displaced & Drought Affec
Location: Zam/Pebane

Contact: Frost
Donor Project No:

Carry Overs 92/93:	
Committments 93/94:	44,286
Total:	44,286

Disbursements or Obligated to Disburse
(Usually means a signed contract)

Estimate if necessary)

Amount: 44,286

Note: To update, just write on a photocopy and send the changes to us. Please, fill in "Obligated to Disburse", "Location" details, etc where missing. Thank you.

Remarks:

(Include any lines which will help the reader to understand the commitment information)

ACTION AID

Activity Number: 4.2.05 Reference No: 0094
Category: RESTORATION OF ESSENTIAL SERVICES
Sector: Health
Title: Rural Health Care
Starting From: 1 January 1993 To: 31 December 1993
Implementing Agency: ACTION AID

Beneficiaries Number:
Classification: Disp/Returnees
Location: Zam/Maganja/Pebane

Contact: Frost
Donor Project No:

Carry Overs 92/93:	
Committments 93/94:	125,000
Total:	125,000

Disbursements or Obligated to Disburse
(Usually means a signed contract)

(Estimate if necessary)

Amount: 59,433

Note: To update, just write on a photocopy and send the changes to us. Please, fill in "Obligated to Disburse", "Location" details, etc where missing. Thank you.

Remarks: Modification 19/07/93

(Include any lines which will help the reader to understand the commitment information)

Activity Number	Donor	Reference Number	Implementing Agency	Carryover 92/93	Commitments 93/94	Obligated to Disburse	% of Needs
	NETHERLANDS	0335	UNDP(TF)/IOM		1,132,161	1,132,161	
	ONUMOZ	0357	IOM		3,500,000	3,500,000	
	SWEDEN	0633	IOM		1,453,500	1,453,500	
	USA	0011	IOM		7,000,000	2,000,000	
	Activity Needs:	15,088,210	Activity Totals:		13,285,661	8,285,661	88 %
2.0.08	Demobilization Subsidies						
	MOZAMBIQUE	0338	TECH UNIT		22,470,000	14,600,000	
	Activity Needs:	22,470,000	Activity Totals:		22,470,000	14,600,000	100 %
2.0.09	Technical Unit for Demobilization						
	EEC	0340	TECH UNIT		216,000	216,000	
	ONUMOZ	0341	TECH UNIT		147,747	0	
	SWEDEN	0632	TECH UNIT		685,000	685,000	
	SWITZERLAND	0339	TECH UNIT		180,000	180,000	
	UNICEF	0342	TECH UNIT		84,000	84,000	
	Activity Needs:	1,312,747	Activity Totals:		1,312,747	1,165,000	100 %
2.0.10	Information & Social Reintegrat.						
	ONUMOZ	0343	TECH UNIT		477,800	89,604	
	Activity Needs:	477,800	Activity Totals:		477,800	89,604	100 %
2.0.11	Demob & Reintegrat. of Vulnerable Groups						
	USA	0626	TECH UNIT		665,000	665,000	
	USA	0650	SCF(USA)		860,000	0	
	Activity Needs:	1,525,000	Activity Totals:		1,525,000	665,000	100 %
2.0.99	Unallocated (Reserved for Reintegration						
	DENMARK	0365	UNOHAC(TF)		1,515,000	0	
	NETHERLANDS	0283	UNDP/OPS		3,278,690	0	
	SWITZERLAND	0345	UNOHAC(TF)/TBD		1,113,950	0	
	Activity Needs:		Activity Totals:		5,907,640	0	No Needs
3.0.00	Allocated to General Project-Emergency						
	ITALY	0510	TO BE DETERMINED		276,450	0	
	JAPAN	0414	ICRC		1,360,000	1,360,000	
	NORWAY	0473	JESUS ALIVE MISSION		1,472,000	1,472,000	
	NORWAY	0474	NRC/ICRC		572,836	572,836	
3.0.99	Unallocated (Reserved for Emerg.Relief)						
	NORWAY	0608	NRC/ICRC-CVM		285,714	285,714	
	Activity Needs:		Activity Totals:		285,714	285,714	No Needs
3.1.00	Allocated to Relief Food						
	GERMANY	0376	GTZ		4,925,000	0	
	NETHERLANDS	0696	MSF(FRANCE)		412,650	322,650	

Activity Number	Donor	Reference Number	Implementing Agency	Carryover 92/93	Commitments 93/94	Obligated to Disburse	% of Needs
1.0.01	Repatriation Operation						
	UNHCR	0002		3,400,000	2,000,000	5,400,000	
	Activity Needs:	11,000,000	Activity Totals:	3,400,000	2,000,000	5,400,000	49 %
	Allocated to a General Project in Demob						
2.0.01	Supply of Food For Demobilization						
	ONUMOZ	0535 WFP		0	350,294	350,294	
	WFP	0381 WFP		0	7,566,514	7,566,514	
	Activity Needs:	7,916,808	Activity Totals:	0	7,916,808	7,916,808	100 %
2.0.02	Supply of Non-food items for Demobilizat						
	ONUMOZ	0349 TECH UNIT		0	2,000,000	876,398	
	SWEDEN	0634		0	261,500	261,500	
	USA	0627 ONOMOZ		0	900,000	277,540	
	Activity Needs:	3,161,500	Activity Totals:	0	3,161,500	1,415,438	100 %
2.0.03	Health Care for Demobilization						
	NORWAY	0350 TECH UNIT/WHO		0	461,538	461,538	
	ONUMOZ	0352 TECH UNIT/WHO		0	2,403,000	1,513,146	
	UNICEF	0351 TECH UNIT/WHO		0	398,651	398,651	
	USA	0169 WHO		0	545,000	545,000	
	WHO	0567 TECH UNIT		0	137,833	137,833	
	Activity Needs:	3,976,022	Activity Totals:	0	3,946,022	3,056,168	99 %
2.0.04	Water Supply and Sanitation for Demobil.						
	CANADA	0695 MSF(FRANCE)		0	36,300	25,700	
	EEC	0353 TECH UNIT		0	850,000	850,000	
	NETHERLANDS	0694 MSF(FRANCE)		33,500	66,500	100,000	
	SOUTH AFRICA	0708 S.A.TRADE MISSION		0	434,782	0	
	Activity Needs:	753,700	Activity Totals:	33,500	1,387,582	975,700	189 %
2.0.05	Civilian Clothing for Demobilization						
	DENMARK	0344 UNDP(TF)/TECH UNIT		0	469,704	469,704	
	NETHERLANDS	0354 UNDP(TF)/TU		0	430,824	430,824	
	ONUMOZ	0355 TECH UNIT		0	1,821,159	533,320	
	Activity Needs:	2,721,687	Activity Totals:	0	2,721,687	1,433,848	100 %
2.0.06	Registration of Demobilizing Soldiers						
	ONUMOZ	0356 TECH UNIT		0	300,000	17,384	
	Activity Needs:	300,000	Activity Totals:	0	300,000	17,384	100 %
2.0.07	Home Transportation for Demobilized						
	IOM	0537 IOM		0	200,000	200,000	

United Nations Office for Humanitarian Assistance Coordination

Committments to Consolidated Humanitarian Assistance Programme
May 1993 - April 1994

BY ACTIVITY

REPORT SUMMARY TOTALS

Needs:	609,688,760
Committments:	549,084,801
Obligated to Disburse:	360,946,515

Balance:	60,603,959	Needs Filled :	90 %

Note: Commitments = 92/93 Carryover + 93/94 Committments

Activity: 3.1.03	**Nutritional Rehabilitation and Hospital Diet**
Target Population:	Children admitted to nutritional rehabilitation centres and hospital patients
Concerned Agencies:	MOH, WHO, UNICEF, Bilateral Donors, NGOs
Time Frame:	May 1993 to April 1994
Objective:	Assure a regular supply of commodities for nutritional rehabilitation. Improve the hospital diet in key health facilities.

Summary

Severe malnutrition has been reported country-wide, particularly in the most drought-affected provinces. In newly accessible areas, information on health and nutrition is scarce due to the lack of basic health services. Some missions report instances of serious malnutrition; this is confirmed by the poor condition of people moving to accommodation centres.

Nutritional rehabilitation, targeting malnourished children, is carried out by the Ministry of Health and NGOs using a mixture of milk, oil and sugar. The need for commodities for therapeutic feeding is increasing and new nutritional rehabilitation centres are being established wherever there are trained personnel.

With the improvement of accessibility and communication, provincial stockpiles of milk, oil and sugar must be created, within the framework of enhanced support to provincial health authorities (4.2.03). The commodities will be delivered to peripheral health units according to prevailing nutritional conditions, the capacity of the health network, and logistics. District hospitals have limited funds for recurring costs, so that food for in-patients is generally insufficient. Chronic patients have special requirements which cannot be met by the current hospital diet. With the expected expansion of the health network these difficulties are likely to worsen. Basic needs for improving the hospital diet include maize, pulses, oil, sugar and milk.

FINANCIAL SUMMARY - US$	
Commodities for hospital diet	2,500,000
Commodities for therapeutic feeding	7,190,000
Total Needs	**9,690,000**
Donors: ADRA GERMANY SWEDEN ITALY/USA EEC NATCOM/UNICEF WFP NORWAY CANADA/WORLD VISION EEC	
Carryover 92/93	1,959,684
Commitments	12,774,085
Total Available and Committed	**14,733,769**
of which 9,356,469 obligated to disburse	64 %
Unmet Needs	**(5,043,769)**

246

Activity: 3.2.01	**Road, Rail and Sea Transport**
Target Population:	1.8 million displaced, affected and returned populations in all provinces
Concerned Agencies:	WFP,UNHCR, Bilateral Donors, NGOs, DPCCN
Time Frame:	May 1993 to April 1994
Objective:	Guarantee regular, timely and adequate delivery of relief assistance to beneficiaries.

Summary

For the coming marketing year, in view of the large amounts of relief assistance required for distribution and the territorial extension of relief activities, transport logistics remains a major challenge. The opening up of new areas and the expected influx of refugees from neighbouring countries require special efforts. Due to the scale, complexity and urgency of aid operations, NGO's and UNILOG are complementing and assisting Government efforts. In the past, internal transport of relief assistance was often determined by the prevailing security situation and the availability of military escorts to accompany food convoys.

Since the signature of the peace accord the situation has improved but logistics remains a vital concern for the continuous provision of relief assistance. The provision of ITSH funding (Internal Transport, Shipping and Handling), is essential if food commodities are to be moved by road, rail and sea to needy areas, and to allow for the contracting of private transporters. NGOs have been working on distribution in Renamo areas and returnee districts to distribute part of the total of 336,000 tons planned for distribution during the 12-month period. DPCCN will assume responsibility for 40% of deliveries, NGOs-30%, and WFP/UNILOG-30%.

FINANCIAL SUMMARY - US$	
Road, rail and sea transport	41,576,769
Total Needs	**41,576,769**
Donors: WFP/USA/GAA/EEC/GERMANY/SWEDEN/UK/SOUTH AFRICA	
Carryover 92/93	6,047,059
Commitments	21,555,400
Total Available and Committed	**27,602,459**
of which 18,475,714 obligated to disburse	67 %
Unmet Needs	**13,974,310**

Activity: 3.2.02	**Airlift Operations**
Target Population:	Affected populations in unaccessible areas
Concerned Agencies:	WFP, UNHCR, Bilateral Donors, NGOs, DPCCN
Time Frame:	May 1993 to April 1994
Objective:	Provide basic relief aid to areas not accessible by other means.

Summary

There is a continued need for airlift operations to various destinations not accessible by any other means, pending the initiation of demining activities and the rehabilitation and reconstruction of many access routes. Airlift operations significantly decreased since May as roads, particularly in Sofala to Inhaminga and Caia, opened up.

With the onset of the rainy season, when secondary and tertiary roads become impassable, some of the accessible locations will, once again, become isolated. During this period, airlifts will constitute the only alternative for delivery of basic supplies. For this purpose, it is estimated that approximately 5,000 mt of relief food commodities will still have to be airlifted between May 1993 and April 1994, at a total estimated cost of USD 2.5 million.

FINANCIAL SUMMARY - US$	
Airlift operations	2,500,000
Total Needs	**2,500,000**
Donors: USA ITALY WFP WVIUSA Carryover 92/93 Commitments	 0 3,833,883
Total Available and Committed of which 833,883 obligated to disburse	**3,833,883** 22 %
Unmet Needs	**(1,333,883)**

Activity: 3.2.03 **Maintenance and Storage**

Target Population: 1.8 million displaced, affected and returned populations in all provinces

Concerned Agencies: WFP, UNHCR, Bilateral Donors, NGOs, DPCCN

Time Frame: May 1993 to April 1994

Objective: Provide spare parts, lubricants and fuel for the DPCCN fleet. Fund additional storage capacity for relief food aid.

Summary

DPCCN remains a major operator for the transportation of relief supplies, despite the increasing involvment of the major private sector, of UNILOG, and of the NGO's. However, the average age of the DPCCN fleet will be over six years. An adequate supply of spare parts becomes even more important as the trucks continue to age. Part of the requirements in spares, fuel and lubricants will be met through ITSH funds, but for those contributions not accompanied by these funds, additional funding has to be made available.

The same reasoning applies to storage costs. In the past, DPCCN was able to make use of relatively adequate storage facilities at low cost. Presently, a large part of the warehouses being used are being returned to their owners. Additional capacity has to be found in locations previously not accessible since the quantity of relief aid is significant. Some tent-type warehouses and roofing materials will also be required.

FINANCIAL SUMMARY - US$	
Fuel and lubricants	768,000
Spares	600,000
Storage	2,000,000
Total Needs	**3,368,000**
Donors: SWEDEN/GAA/E 'GERMANY	
Carryover 92/	1,311,039
Commitments	3,911,095
Total Available and Committed	**5,222,134**
of which 5,222,134 obligated to disburse	100 %
Unmet Needs	**(1,854,134)**

Activity: 3.3.01 **Internal Transportation of Displaced Vulnerable Groups**

Target Population: Estimated 250,000 persons within Mozambique

Concerned Agencies: CENE, Social Welfare, NAR, Bilateral Donors, IOM

Time Frame: August 1993 to July 1994

Objective: Provide transport assistance to vulnerable groups among the internally displaced.

Summary

An estimated 4 million internally displaced people will re-settle within Mozambique as a result of the peace process. Of these, many are very far from their home villages and will be unable to reach their destinations unassisted. Some of the aged, sick, handicapped, and those weakened by hunger will also be unable to walk far, or afford to pay for public transportation.

The International Organization for Migration (IOM) has identified a number of partners in each province and has researched where these vulnerable groups are. The registration process is now well underway. Liaison is taking place with communities to involve them in the selection and planning processes. Names of those selected are listed, along with their desired time of return, eligibility category, point of origin, destination, and if they will plant. Numbers are kept centrally by place of origin and destination. By the end of October 1993, this project will already assist 7,000 vulnerable persons to reach their destinations, mainly with IOM-organized transport.

The transport system uses the available mix of transport in each province, contracting a basic fleet and hiring other small transporters as needed. An estimated 12,500 are being given a self-resettlement grant. For 237,500 travelling by organized transport, food rations for resettlement are being issued at the journey's beginning, and food or a travel subsidy is being provided en route. The transport of large groups to a single area at one time is being avoided, in preference for gradual arrivals. Local authorities, national agencies and NGOs are informed of cases requiring special assistance, and of the schedule for arrivals.

FINANCIAL SUMMARY - US$	
Office costs Operational costs Staff	1,024,000 5,225,000 1,145,468
Total Needs	**7,394,468**
Donors: EEC/NORWAY/DENMARK/SWEDEN Carryover 92/93 Commitments	 0 2,107,692
Total Available and Committed of which 707,692 obligated to disburse	**2,107,692** 34 %
Unmet Needs	**5,286,776**

Activity: 3.4.01 **Relief and Survival Items for Returnees and Internally Displaced**

Target Population: Eligible beneficiaries among returnees and internally displaced.

Concerned Agencies: DPCCN, NAR, UNHCR, DHA, NGOs

Time Frame: May 1993 to April 1994

Objective: Provide essential survival and relief items to the most needy people.

Summary

It is estimated that 300.000 internally displaced Mozambicans will require distributions of relief and survival items from the Consolidated Humanitarian Assistance Programme. Each family of five members will receive one set of clothing (one extra length of cloth for women), an average of 2.5 blankets, one cooking set, and 6 kilogrammes of soap (500 grammes per month). The total cost of this package per family will be USD 69.50.

Some returnees will also be targeted. Those returning from neighbouring countries received assistance while in the countries of asylum and it is expected that they will travel with these items when they return to their areas of origin. However, those refugees who were not in camps will be treated as displaced persons. Within Mozambique UNHCR will distribute plastic sheeting to returnees who for various reasons, are unable to roof their houses, as well as buckets for carrying water. For vulnerable groups such as pregnant women, the aged and others, some tents will be distributed until homes can be built.

To date, most of the support for relief and survival items has been provided by NGOs - both national and international - to those internally displaced families living in areas which were inaccessible during the war years. These areas and these families are priorities. UNOHAC coordinates distribution of items to Renamo areas, and since May 1993 several agencies have distributed items to these areas.

FINANCIAL SUMMARY - US$	
Logistics and storage	1,146,750
Purchase of relief and survival items	4,587,000
Purchase of water jars, tents and plastic sheeting	2,000,000
Total Needs	**7,733,750**
Donors: UNHCR/DHA/SCF(UK)/FHI//EEC/GERMANY/SOUTH APRICA/DENMARK WORLD VISION/CHINA	
Carryover 92/93	809,892
Commitments	2,965,431
Total Available and Committed	3,775,323
of which 1,748,652 obligated to disburse	46 %
Unmet Needs	**3,958,427**

Activity: 3.4.02 **Emergency Stockpile Project**

Target Population: Eligible beneficiaries among returnees, internally displaced and victims of natural disasters

Concerned Agencies: CENE, DPCCN, DHA, NGOs

Time Frame: May 1993 to April 1994

Objective: Respond quickly to the unforeseen emergency situation for the most needy people.

Summary

The stockpile warehouses in Maputo, Beira and Quelimane enable the stockpile project to provide and distribute, on short notice: clothing, blankets, cooking utensils, soap, shelter, a water supply, and tools for sanitation. These go to the most needy people not otherwise assisted. In case of emergency the stockpile can provide these relief items for distribution through NGOs. To shorten the delivery time to the districts and reduce the internal transport costs, the construction of 6 new equipped, prefabricated warehouses is planned (Quelimane for 100,000 people, Inhambane, Chimoio, Tete, Morrumbala and Nampula for 50,000 people each). Survival and relief items should also be provided for Maputo (50,000) and Beira (100,000 people).

This project started 5 years ago and is operating smoothly. The stockpile has been particularly important in ensuring distribution to previously inaccessible areas, particularly Renamo zones. Its continuation on an expanded basis is essential to respond to the needs of the most vulnerable families.

FINANCIAL SUMMARY - US$	
Logistics and distribution	2,077,500
New equipped warehouses	595,000
Purchase of relief items	8,310,000
Technical assistance and operation costs	828,800
Total Needs	**11,811,300**
Donors: DHA/SWEDEN/EEC	
Carryover 92/93	0
Commitments	3,064,222
Total Available and Committed	**3,064,222**
of which 1,264,222 obligated to disburse	41 %
Unmet Needs	**8,747,078**

CABO DELGADO Province

MECUFI District

ARO

4.2.04 Rehabilitation of Health Network
From: 01 01 93 **To:** 31 12 94
Location: MECUFI
— Construction Of Mecufi Health Center

GAA

3.1.01 Food Relief Distributions
From: 01 04 93 **To:** 31/07/93
— Food Relief Distribution

3.1.01 Food Relief Distributions
From: 01 04 93 **To:** 31 07 93
Location: MECUFI
— Food Dist.-Displaced & Returnees Incl.renamo Area

3.2.01 Road, Rail and Sea Transport
From: 01 04 93 **To:** 31 07 93
— Food Aid Logistics

3.2.03 Maintenance and Storage
— Maintenance And Storage

3.4.01 Returnees and Internally Displaced
— Returnees And Internally Displaced

4.1.01 Seeds & Tools (see 4.1.02 & 4.1.03)
— Seeds For Family Sector

4.1.02 Seed for Family Sector
From: 01 04 93 **To:** 31/07/93
— Agricultural Tools

MARINS SAN FRONTIERE

4.2.05 Rural Health Care
Location: MECUFI
— Health Assis.in Costal Areas. Assis Provided-Boat

SCF(USA)

4.6.07
Location: MECUFI
— Children & War Proj.- Suport To War Affec.children

MELUCO District

CRIAA

4.6.01 Area Based Programmes at Community Level
Location: MELUCO

SCF(USA)

4.6.07
Location: MELUCO
— Children & War Proj.- Suport To War Affec.children

MOCIMBOA DA PRAIA District

CRIAA

4.6.01 Area Based Programmes at Community Level
Location: MOCIMBOA DA PRAIA

MARINS SAN FRONTIERE

4.2.05 Rural Health Care
Location:
— Health Assis.in Costal Areas. Assis Provided-Boat

CABO DELGADO Province

SCF(USA)
4.6.07 — Location: ANCUABE — Children & War Proj.- Suport To War Affec.children

4.6.07 — Location: — Children & War Proj.- Suport To War Affec.children

BALAMA District

GAA

3.1.01	Food Relief Distributions From: 01 04 93 To: 31 07 93		Food Relief Distribution
3.1.01	Food Relief Distributions From: 01 04 93 To: 31 07 93 Location: BALAMA		Food Dist.-Displaced & Returnees Incl.renamo Area
3.2.01	Road, Rail and Sea Transport From: 01 04 93 To: 31 07 93		Food Aid Logistics
3.2.03	Maintenance and Storage		Maintenance And Storage
3.4.01	Returnees and Internally Displaced		Returnees And Internally Displaced
4.1.01	Seeds & Tools (see 4.1.02 & 4.1.03)		Seeds For Family Sector
4.1.02	Seed for Family Sector From: 01 04 93 To: 31 07 93		Agricultural Tools

SCF(USA)
4.6.07 — Location: BALAMA — Children & War Proj.- Suport To War Affec.children

CHIURE District

GAA

3.1.01	Food Relief Distributions From: 01/04 93 To: 31/07/93		Food Relief Distribution
3.1.01	Food Relief Distributions From: 01/04/93 To: 31/07/93 Location: CHIURE		Food Dist.-Displaced & Returnees Incl.renamo Area
3.2.01	Road, Rail and Sea Transport From: 01/04/93 To: 31/07/93		Food Aid Logistics
3.2.03	Maintenance and Storage		Maintenance And Storage
3.4.01	Returnees and Internally Displaced		Returnees And Internally Displaced
4.1.01	Seeds & Tools (see 4.1.02 & 4.1.03)		Seeds For Family Sector
4.1.02	Seed for Family Sector From: 01/04/93 To: 31/07/93		Agricultural Tools

SCF(USA)
4.6.07 — Location: CHIURE — Children & War Proj.- Suport To War Affec.children

MACOMIA District

MARINS SAN FRONTIERE
4.2.05 Rural Health Care — Location: MACOMIA — Health Assis.in Costal Areas. Assis Provided-Boat

SCF(USA)
4.6.07 — Location: MACOMIA — Children & War Proj.- Suport To War Affec.children

4.6.07 — Location: — Children & War Proj.- Suport To War Affec.children

United Nations Office for Humanitarian Assistance Coordination

Shaman: System for Humanitarian Assistance Management

WHO IS DOING WHAT WHERE

Report

Ranked by Province District Agency Activity

UNOHAC SHAMAN

UNOHAC SHAMAN — Who-What-Where Province - Current to: 29 December 1993

Province

IBO District

GAA			
3.1.01	Food Relief Distributions **From:** 01/04/93 **To:** 31/07/93		Food Relief Distribution
3.2.01	Road, Rail and Sea Transport **From:** 01/04/93 **To:** 31/07/93		Food Aid Logistics
3.2.03	Maintenance and Storage		Maintenance And Storage

MANDLAKAZE District

SCF(USA)		
4.1.01	Seeds & Tools (see 4.1.02 & 4.1.03)	Seeds And Tool Distribution

CABO DELGADO Province

ANCUABE District

GAA		
3.1.01	Food Relief Distributions **From:** 01/04/93 **To:** 31/07/93	Food Relief Distribution
3.1.01	Food Relief Distributions **From:** 01/04/93 **To:** 31/07/93 **Location:** ANCUABE	Food Dist.-Displaced & Returnees Incl.renamo Area
3.2.01	Road, Rail and Sea Transport **From:** 01/04/93 **To:** 31/07/93	Food Aid Logistics
3.2.03	Maintenance and Storage	Maintenance And Storage
3.4.01	Returnees and Internally Displaced **Location:** ANCUABE	Returnees And Internally Displaced
4.1.01	Seeds & Tools (see 4.1.02 & 4.1.03)	Seeds For Family Sector
4.1.02	Seed for Family Sector From: 01/04/93 To: 31/07/93	Agricultural Tools

INHAMBANE Province

AIFA			
4.1.01	Seeds & Tools (see 4.1.02 & 4.1.03)	Agricultural Prodiction	RN: [illegible]
4.1.02	Seed for Family Sector	Agricultural Production	RN: [illegible]
4.1.08	Reserve Grain Silo, Beira	Agricultural Production	RN: [illegible]
4.2.00	Allocated to a General Health Project	Small Child Survival - Feeding Programme	RN: [illegible]
4.2.02	Cold Chain Equipment/Vaccines From: 01/03/93 To: 31/12/94	Health Services	RN: [illegible]
4.3.01	Water Supply for Resettlement	Water And Sanitation	RN: [illegible]
4.3.01	Water Supply for Resettlement	Water And Sanitation	RN: [illegible]
4.5.01	Education and Teachers Training	Education	RN: [illegible]

MSF BELGIUM		Description:	
4.2.04	Rehabilitation of Health Network	Rehabilitation Of Health Networks	RN: [illegible]
4.2.05	Rural Health Care	Medical Supervision, Supply Of Drugs & Water Prov.	RN: [illegible]

JANGAMO District

ACTION NORD-SUD		Description:	
4.1.00	Allocated to an Agriculture Project From: 01/01/93 To:	Construction Of 3 Dip Tanks	RN: [illegible]
4.2.04	Rehabilitation of Health Network From: 01/01/93 To:	Rehab. Of 5 Health Posts In Coop.with Amdu	RN: [illegible]
4.5.01	Education and Teachers Training From: 01/01/91 To:	Rehab. Of 90 Classrooms In Coop.with Amdu	RN: [illegible]
6.0.00	Allocated to General Institut. Support From: 01/01/93 To: 31/12/94 Location: INHAMBANE	Study On Development Needs For Inhambane Province	RN: [illegible]

ASTER		Description:	
4.6.00	Alloc. to a General Multi-Sectoral Prog. Location: JANGAMO	Tech.assistance,construc.material,medic.tech,rehab	RN: [illegible]

CCM		Description:	
4.6.01	Area Based Programmes at Community Level Location: MUTAMBA	Joint Project With Lwf	RN: [illegible]

COMMUNITY AID ABROAD		Description:	
4.1.00	Allocated to an Agriculture Project Location: INHAMBANE	Agricultural Project In Inhambane Free Zones	RN: [illegible]
4.2.04	Rehabilitation of Health Network	Construction Of First Aid Post	RN: [illegible]
4.3.01	Water Supply for Resettlement	Water Supply To Maunza Displaced Camp	RN: [illegible]
4.5.01	Education and Teachers Training	Improvement Of 4 Classrooms In Maunza Camp.	RN: [illegible]

CROCEVIA		Description:	
4.5.00	Allocated to a general Education Project	Laboratories In Secondry Schools	RN: [illegible]

KULIMA		Description:	
4.1.00	Allocated to an Agriculture Project Location: INHAMBANE	Support To Rural Communities	RN: [illegible]
4.1.00	Allocated to an Agriculture Project Location: MAXIXE	Support To Rural Communities	RN: [illegible]

256

HAMBANE Province

ADRA			
4.2.00	Allocated to a General Health Project	Description: Small Child Survival - Feeding Programme	RN: [illegible]
4.2.02	Cold Chain Equipment Vaccines From: 01/03/93 To: 28/02/94	Health Services	RN: [illegible]
4.3.01	Water Supply for Resettlement	Water And Sanitation	RN: [illegible]
4.3.01	Water Supply for Resettlement	Water And Sanitation	RN: [illegible]
4.5.01	Education and Teachers Training	Education	RN: [illegible]
MSF BELGIUM		Description:	
4.2.04	Rehabilitation of Health Network	Rehabilitation Of Health Networks	RN: [illegible]
4.2.05	Rural Health Care	Medical Supervision, Supply Of Drugs & Water Prov	RN: [illegible]
RRR PROJECT		Description:	
4.6.00	Alloc. to a General Multi-Sectoral Prog. From: 01/01/93 To:		RN: [illegible]

HOMOINE District

CCM		Description:	
4.6.01	Area Based Programmes at Community Level Location: AGOSTINO NETO	Joint Proj with Lwf-Agri-Heal-Edu-Incom.generated	RN: [illegible]
LWF		Description:	
4.6.01	Area Based Programmes at Community Level Location: AGOSTINO NETO	Joint Project With Ccm	RN: [illegible]

INHARRIME District

ACTION NORD-SUD		Description:	
4.1.00	Allocated to an Agriculture Project From: 01/01/93 To:	Construction Of 3 Dip Tanks	RN: 00884
4.2.04	Rehabilitation of Health Network From: 01/01/93 To:	Rehab. Of 5 Health Posts In Coop.with Amdu	RN: 00880
4.5.01	Education and Teachers Training From: 01/01/93 To:	Rehab. Of 90 Classrooms In Coop.with Amdu	RN: 00876
ASTER		Description:	
4.6.00	Alloc. to a General Multi-Sectoral Prog. Location: INHARRIME	Tech.assistance,construc.material,medic.tech,rehab	RN: 01125
KULIMA		Description:	
4.6.01	Area Based Programmes at Community Level From: 01/05/93 To: 31/12/95 Location: INHARRIME	Intergrated Development Programme	RN: 00868

INHASSORO District

ADRA		Description:	
1.0.01	Repatriation Operation From: 01/03/93 To: 28/02/94	Helping Returnees	RN: 00061
3.1.01	Food Relief Distributions	Emergency Relief	RN: 00092
3.1.02	Food for Work	Food For Work Programmes	RN: [illegible]
3.1.02	Food for Work	Rehab Of Schools,health Posts,roads & Water Resor	RN: [illegible]
3.1.03	Nutritional Rehabilitation	Supplementary Feeding	RN: [illegible]
4.1.00	Allocated to an Agriculture Project	Training Of Farmers And Provision Of Seeds & Tools	RN: [illegible]

United Nations Office for Humanitarian Assistance Coordination
SHAMAN: System for Humanitarian Assistance Management

WHO-WHAT-WHERE/PROVINCE

Current to: 29 December 1993

Agency Activities by Province by District by Agency
Ranked by Province by District by Agency by Activity

INHAMBANE Province

FUNHALOURO District

ADRA		Description:	
1.0.01	Repatriation Operation From: 01/03/93 To: 28/02/94	Helping Returnees	RN: 00059
3.1.01	Food Relief Distributions	Emergency Relief	RN: 00090
3.1.02	Food for Work	Food For Work Programmes	RN: 00085
3.1.03	Nutritional Rehabilitation	Supplementary Feeding	RN: 00080
4.1.01	Seeds & Tools (see 4.1.02 & 4.1.03)	Agricultural Production	RN: 00075
4.1.02	Seed for Family Sector	Agricultural Production	RN: 00070
4.1.04	Reserve Grain Silo, Beira	Agricultural Production	RN: 00064
4.2.02	Cold Chain Equipment/Vaccines From: 01/03/93 To: 28/02/94	Health Services	RN: 00053
4.3.01	Water Supply for Resettlement	Water And Sanitation	RN: 00043
4.3.01	Water Supply for Resettlement	Water And Sanitation	RN: 00048
4.5.01	Education and Teachers Training	Education	RN: 00038
ASTER		Description:	
4.6.00	Alloc. to a General Multi-Sectoral Prog. Location: FUNHALOURO	Tech.assistance,construc.material,medic.tech,rehab	RN: 01124
CARE		Description:	
4.1.01	Seeds & Tools (see 4.1.02 & 4.1.03)		RN: 00916

GOVURO District

ADRA		Description:	
1.0.01	Repatriation Operation From: 01/03/93 To: 28/02/94	Helping Returnees	RN: 00060
3.1.01	Food Relief Distributions	Emergency Relief	RN: 00091
3.1.01	Food Relief Distributions	Food Distribution - 13,000 Beneficiaries	RN: 00639
3.1.02	Food for Work	Food For Work Programmes	RN: 00086
3.1.02	Food for Work	Rehab Of Schools,health Posts,roads & Water Resor.	RN: 00649
3.1.03	Nutritional Rehabilitation	Supplementary Feeding	RN: 00081
4.1.02	Allocated to an Agriculture Project	Training Of Farmers And Provision Of Seeds & Tools	RN: 00641
4.1.01	Seeds & Tools (see 4.1.02 & 4.1.03)	Agricultural Production	RN: 00076
4.1.02	Seed for Family Sector	Agricultural Production	RN: 00071
4.1.04	Reserve Grain Silo, Beira	Agricultural Production	RN: 00066

298

ADRA

Contact: Taylor
Donor Project No:

Activity Number: 3.1.03 Reference No: 0037
Category: EMERGENCY RELIEF
Sector: Relief Food Aid
Title: Nutritional Rehabilitation
Starting From: 1 October 1992 To: 31 October 1993
Implementing Agency: ADRA

Carry Overs 92/93:	
Committments 93/94:	35,000
Total:	35,000

Beneficiaries Number: 13,300
Classification: Children & Mothers
Location: Inh/Sof

Disbursements or Obligated to Disburse
(Usually means a signed contract)

(Estimate if necessary)

Amount: 26,250

Note: To update, just write on a photocopy and send the changes to us. Please, fill in "Obligated to Disburse", "Location" details, etc where missing. Thank you.

Remarks:

(Include any lines which will help the reader to understand the commitment information)

ADRA

Contact: Taylor
Donor Project No:

Activity Number: 3.4.99 Reference No: 0048
Category: EMERGENCY RELIEF
Sector: Non-Food Relief
Title: Unallocated (Reserved for Non-Food)
Starting From: 1 October 1992 To: 30 September 1993
Implementing Agency: ADRA

Carry Overs 92/93:	
Committments 93/94:	400,000
Total:	400,000

Beneficiaries Number:
Classification: Displaced & Returnees
Location: Country Wide

Disbursements or Obligated to Disburse
(Usually means a signed contract)

(Estimate if necessary)

Amount: 320,000

Note: To update, just write on a photocopy and send the changes to us. Please, fill in "Obligated to Disburse", "Location" details, etc where missing. Thank you.

Remarks:

(Include any lines which will help the reader to understand the commitment information)

124	HOME_8	Character	10	Relation 8 Set From this Table
125	RLKEY_8	Character	40	Relational Expression 8
126	LUKUP_8	Character	10	Relation 8 Set Into this Table
127	RLTYPE_8	Character	1	Relation Type, "P", "C" or ""
128	HOME_9	Character	10	Relation 9 Set From this Table
129	RLKEY_9	Character	40	Relational Expression 9
130	LUKUP_9	Character	10	Relation 9 Set Into this Table
131	RLTYPE_9	Character	1	Relation Type, "P", "C" or ""
132	HOME_10	Character	10	Relation 10 Set From this Table
133	RLKEY_10	Character	40	Relational Expression 10
134	LUKUP_10	Character	10	Relation 10 Set Into this Table
135	RLTYPE_10	Character	1	Relation Type, "P", "C" or ""
			2410	

Index Files PK - SHA_DIC1 = ALIAS

SHAMAN.MNU (SHAMAN Menu System Table)

Field	Field Name	Type	Width	Dec	Description
1	MENU_ID	Character	15		ID Number of Menu
2	PROMPT	Character	40		Menu Prompt
3	CHILD_ID	Character	15		
4	BLOCK	Character	50		Block for Executable
5	MESSAGE	Character	80		One-line Message to User
6	HELP_TEXT	Memo	10		Long Help Text
7	SPECS	Memo	10		Specifications
8	SECURITY	Character	10		Security Level
9	EXECUTE	Logical	1		Execute this Menu Item?
			232		

Index Files PK - SHAMAN1 = MENU_ID

SHA_QRY (Query Manager Details)

Field	Field Name	Type	Width	Dec	Description
1	QUERY_NAME	Character	25		Used in Query Read Menu
2	DATA_FILE	Character	8		Controlling DBF
3	USERNUM	Character	4		User # (Compared with Sys)
4	OPEN	Logical	1		Can Another User Modify?
5	QUERY	Memo	10		Query Specification
			49		

Index Files PK - SHA_QRY1 = DATA_FILE + QUERY_NAME

SHA_SYS (System Details)

Field	Field Name	Type	Width	Dec	Description
1	USERNAME	Character	40		User Name
2	USERNUM	Character	4		User Number
3	PASSWORD	Character	8		Password
4	PERMISSION	Character	10		EXPERT is highest
5	USERPATH	Character	40		Not Presently Used
6	SCR_PATH	Character	20		RAMdisk Drivespec if any
7	CLOSE_OK	Logical	1		
8	EDIT_HELP	Logical	1		Allowed to Edit Help?
9	COLOR_1	Character	20		SysColor() Definitions
10	COLOR_2	Character	20		SysColor() Definitions
11	COLOR_3	Character	20		SysColor() Definitions
12	COLOR_4	Character	20		SysColor() Definitions
13	COLOR_5	Character	20		SysColor() Definitions
14	COLOR_6	Character	20		SysColor() Definitions
15	COLOR_7	Character	20		SysColor() Definitions
16	COLOR_8	Character	20		SysColor() Definitions
17	COLOR_9	Character	20		SysColor() Definitions
18	COLOR_10	Character	20		SysColor() Definitions

19	COLOR_11	Character	20	SysColor() Definitions
20	COLOR_12	Character	20	SysColor() Definitions
21	COLOR_13	Character	20	SysColor() Definitions
22	MONO_1	Character	20	SysColor() Definitions
23	MONO_2	Character	20	SysColor() Definitions
24	MONO_3	Character	20	SysColor() Definitions
25	MONO_4	Character	20	SysColor() Definitions
26	MONO_5	Character	20	SysColor() Definitions
27	MONO_6	Character	20	SysColor() Definitions
28	MONO_7	Character	20	SysColor() Definitions
29	MONO_8	Character	20	SysColor() Definitions
30	MONO_9	Character	20	SysColor() Definitions
31	MONO_10	Character	20	SysColor() Definitions
32	MONO_11	Character	20	SysColor() Definitions
33	MONO_12	Character	20	SysColor() Definitions
34	MONO_13	Character	20	SysColor() Definitions
			645	

Index Files PK - SHA_SYS1 USERNUM

SHA_HELP (System Help)

Field	Field Name	Type	Width	Dec	Description
1	SUBJECT	Character	35		Title to Present in Help Index
2	HELP	Memo	10		Help Text
3	HELP_KEY	Character	15		Key Field (SEEK HelpCode())
4	VARIABLE	Character	15		READ Variable if any
5	HTR	Numeric	2		Top Row
6	HTC	Numeric	2		Top Column
7	HBR	Numeric	2		Bottom Row
8	HBC	Numeric	2		Bottom Column
			84		

Index Files PK - SHA_HEL1 - HELP_KEY + VARIABLE

PRINTER (Printer Details)

Field	Field Name	Type	Width	Dec	Description
PK 1	TYPE_STYLE	Character	20		Key Field - SEEK Code
2	CHR_1	Numeric	3		ASCII Char to Send
3	CHR_2	Numeric	3		ASCII Char to Send
4	CHR_3	Numeric	3		ASCII Char to Send
5	CHR_4	Numeric	3		ASCII Char to Send
6	CHR_5	Numeric	3		ASCII Char to Send
7	CHR_6	Numeric	3		ASCII Char to Send
8	CHR_7	Numeric	3		ASCII Char to Send
9	CHR_8	Numeric	3		ASCII Char to Send
10	CHR_9	Numeric	3		ASCII Char to Send
11	CHR_10	Numeric	3		ASCII Char to Send
12	CHR_11	Numeric	3		ASCII Char to Send
13	CHR_12	Numeric	3		ASCII Char to Send
			57		

Index Files PK - PRINTER1 - TYPE_STYLE

261

SHA_RPT (Report Manager Details)

Field	Field Name	Type	Width	Dec	Description
1	RI_REPORT	Character	30		Name of Report
2	RI_LIBRARY	Character	20		RPT file if any (R&R)
3	RI_MASTER	Character	20		Controlling DBF (Name)
4	RI_ALIAS	Character	10		Controlling DBF Alias
5	RI_ORDER	Numeric	2		Controlling Index
6	RI_KEYEXP	Character	40		Key Expression (Control NTX)
7	RI_KEYCUE	Character	40		Scope Cue
8	RI_KEYPIC	Character	40		Scope Picture
9	RI_OUTFILE	Character	12		Output File if any
10	RI_OUTAPPN	Logical	1		Append if it exists?
11	RI_PEQUIT	Logical	1		Quit on Printer Error?
12	RI_CHKTIME	Character	1		Freq of Error Check (s)
13	RI_NOESC	Logical	1		Allow Escape to Terminate?
14	RI_PRINTER	Character	1		D/isplay, A/scii File, or 1-9
15	RI_BEGPAGE	Numeric	4		Begin at Page (R&R)
16	RI_ENDPAGE	Numeric	4		End at Page (R&R)
17	RI_TEST	Logical	1		Begin with Test Option?
18	RI_SCOPE	Character	1		E/ntire, as S/aved, O/verride
19	RI_LOSCOPE	Character	50		Begin at (Index Key)
20	RI_HISCOPE	Character	50		End at (Index Key)
21	RI_COPIES	Numeric	4		# Copies to Print
22	RI_FILTER	Character	250		Query Expression
23	RI_QUERY	Character	1		E/ntire, as S/aved, O/verride
24	RI_DISPERR	Logical	1		Display Error Messages?
25	RPT_TYPE	Character	1		R/eport Writer, H/ard Coded, Clipper Report F/orm, L/abel
26	RPT_INIT	Character	20		UDF to Execute Before RPT
27	RPT_MAIN	Character	20		UDF - Coded Report
28	RPT_EXIT	Character	20		UDF to Execute After RPT
29	RPT_FRMLBL	Character	8		FRM or LBL Name if any
30	RPT_HEAD	Character	40		FRM Header
31	RPT_SUMM	Logical	1		FRM Summary or Detail
32	IS_FRONT	Logical	1		Invoke Report Options?
			697		

Index Files PK - SHA_RPTI - RI_REPORT

End of SHAMAN Files

Activity Title Authority File

Activity Number	Short / Long Title
1	REPATRIATION OF REFUGEES Repatriation of Refugees
1 0	Repatriation Operation Repatriation Operation
1 0 01	Repatriation Operation Logistical Support to Repatriation Operation
2	DEMOBILIZATION Demobilization
2 0	Demobilization Demobilization
2 0 00	Allocated to a General Project in Demob Allocated to a General Project in Demobilization
2 0 01	Supply of Food For Demobilization Supply of Food for Demobilization
2 0 02	Supply of Non-food Items for Demobilizat Supply of Necessities for Demobilization
2 0 03	Health Care for Demobilization Primary Health Care for Demobilization
2 0 04	Water Supply and Sanitation for Demobil Water Supply and Sanitation for Demobilization
2 0 05	Civilian Clothing for Demobilization Supply of Civilian Clothing for Demobilization
2 0 06	Registration of Demobilizing Soldiers Registration of Demobilizing Soldiers
2.0.07	Home Transportation for Demobilized Home Transportation for Demobilized
2.0.08	Demobilization Subsidies Payment of Demobilization Subsidies
2 0.09	Technical Unit for Demobilization Technical Unit for Demobilization
2.0.10	Information & Social Reintegrat. Information and Social Reintegration.
2 0 11	Demob & Reintegrat of Vulnerable Groups Demobilization and Reintegration of Vulnerable Groups
2.0 99	Unalloc'ed (Reserved for Reintegration Unallocated (Reserved to Reintegration)
3	EMERGENCY RELIEF Emergency Relief
3.0.00	Allocated to General Project-Emergency Allocated to General Project-Emergency
3.0.99	Unallocated (Reserved for Energ.Relief) Unallocated (Reserved for Emergency Relief)
3 1	Relief Food Aid Relief Food Aid
3.1.00	Allocated to Relief Food Allocated to Relief Food
3 1 01	Food Relief Distributions Food Relief Distributions

Activity Number	Short / Long Title
3.1.02	Food for Work
	Food for Work Programmes
3.1.03	Nutritional Rehabilitation
	Nutritional Rehabilitation and Hospital Diet
3.1.99	Unallocated (Reserved for Relief Food)
	Unallocated (Reserved for Relief Food)
3.2	Logistics
	Logistics
3.2.00	Allocated to a General Logistics Project
	Allocated to a General Logistics Project
3.2.01	Road, Rail and Sea Transport
	Road, Rail and Sea Transport
3.2.02	Airlift Operations
	Airlift Operations
3.2.03	Maintenance and Storage
	Maintenance and Storage
3.2.99	Unallocated (Reserved for Logistics)
	Unallocated (Reserved for Logistics)
3.3	Transport of People
	Transport of People
3.3.00	Allocated to General Transport of People
	Allocated to General Transport of People
3.3.01	Transport of Vulnerable Groups
	Internal Transportation of Displaced Vulnerable Groups
3.4	Non-Food Relief
3.4.00	Allocated to General Non-Food Relief
	Allocated to General Non-Food Relief
3.4.01	Returnees and Internally Displaced
	Relief and Survival Items for Returnees and Internally Displaced
3.4.02	Emergency Stockpile Project
	Emergency Stockpile Project
3.4.99	Unallocated (Reserved for Non-Food)
	Unallocated (Reserved for Non-food)
4	RESTORATION OF ESSENTIAL SERVICES
4.0.00	Allocated to Restoration Services
	Allocated to Restoration Services
4.0.99	Unallocated (Reserved for Restor.Serv.)
	Unallocated (Reserved for Restoration of Services)
4.1	Agriculture
4.1.00	Allocated to an Agriculture Project
	Allocated an Agriculture Project
4.1.01	Seeds & Tools (see 4.1.02 & 4.1.03)
	Seeds & Tools (see 4.1.02 & 4.1.03)
4.1.02	Seed for Family Sector
	Supply of Seed and Local Multiplication for Family Sector
4.1.03	Supply of Agricultural Tools
	Supply of Agricultural Tools
4.1.04	Pest Control and Seed Inspection
	Pest Control and Seed Inspection
4.1.05	Cassava Production
	Cassava Multiplication
4.1.06	Marketing of Agricultural Produce
	Marketing of Agricultural Products
4.1.07	Crash Agricultural Extension Programme
	Crash Agricultural Extension Programme
4.1.08	Reserve Grain Silo, Beira
	Reserve Grain Silo, Beira

Activity Number	Short / Long Title
4.1.99	Unallocated (Reserved for Agriculture)
	Unallocated (Reserved for Agriculture)
4.2	Health
4.2.00	Allocated to a General Health Project
	Allocated to a General Health Project
4.2.01	Stockpiles of Drugs/Reagents
	Stockpiles of Emergency Drugs/Reagents
4.2.02	Cold Chain Equipment/Vaccines
	Procurement and Distribution of Vaccines and Cold-Chain Equipment
4.2.03	Support to Provinc. Health Directorates
	Support to Provincial Health Directorates
4.2.04	Rehabilitation of Health Network
	Strengthening and Rehabilitation of the Health Network
4.2.05	Rural Health Care
	Rural Health Care
4.2.06	Prosthetics Assistance
	Medical Assistance and Support to Physically-Disabled Persons
4.2.99	Unallocated (Reserved for Health)
	Unallocated (Reserved for Health)
4.3	Water
	Water
4.3.01	Water Supply for Resettlement
	Water Supply and Sanitation for Rehabilitation and Resettlement
4.4	Roads and Bridges
	Roads and Bridges
4.4.01	Roads and Bridges
	Roads and Bridges
4.5	Education
	Education
4.5.00	Allocated to a general Education Project
	Allocated to a General Education Project
4.5.01	Education and Teachers Training
	Expansion of Basic Education Services and Upgrading of Teachers
4.6	Multi-sectoral Programs
4.6.00	Alloc. to a General Multi-Sectoral Prog.
	Allocated to a General Multisectoral Programme
4.6.01	Area Based Pr grammes at Community Level
	Area-Based P ogrammes at Community Level
4.6.02	Support to Returnee Areas (QIPS)
	Support to Returnee Areas (including QIPS)
4.6.03	Institutional Support for Demob Soldiers
	Institutional Support for Demobilized Soldiers
4.6.04	Training for Demobilized Soldiers
	Training for Demobilized Soldiers
4.6.05	Kits and Credit Scheme for Demobilized S
	Kits and Credit Scheme for Demobilized Soldiers
4.6.06	Employment for Demobilized Soldiers
	Employment for Demobilized Soldiers
4.6.99	Unallocated (Reserved for Multi-Sector)
	Unallocated (Reserved for Multi-Sectoral Programmes)
4.7	Mine Clearance
4.7.00	Allocated to Mine Clearance
	Allocated to Mine-Clearance
4.7.01	Mine Survey & Road Mine Clearance
	Mine Survey and Mine Clearance of Roads

Activity Number	Short / Long Title
* 4.7.02	Mine Awareness & Clearance Training Mine Awareness and Clearance Training
4.7.99	Unallocated (Reserved for Mine Clearance Unallocated (Reserved for Mine-Clearance)
5	BALANCE OF PAYMENTS AND BUDGET SUPPORT
5.0	Balance of Payment Support
5.0.00	Alloc.to general Balance of Paym Support Allocated Balance of Payment and Budget Support
5.0.01	Market Food Aid Market Food Aid
5 0.02	Balance of Payments - Seeds & Tools Balance of Payments Support - Seeds & Tools
5.0.99	Unallocated (Reserv.Balance of Payments) Unallocated (Reserved for Balance of Payments Support)
6	INSTITUTIONAL SUPPORT
6.0	Institutional Support
6.0.00	Allocated to General Institut. Support Allocated to General Institutional Support
6.0.01	Food Security & Early Warning Systems Food Security and Early Warning Systems
6.0.02	Support to CPEs and CDEs Institutional Support for Provincial and District Emergency Commissions
6.0.03	Institutional Support to DPCCN Institutional Support to DPCCN
6.0.04	MOH Emergency Response Support to MOH Emergency Response
6.0.05	National NGOs Support to National Non-Government Organizations
6.0.06	Support to NAR Support to Nucleo de Apoio aos Refugiados (NAR)
6.0.07	Ministry of Education Emergency Response Ministry of Education Emergency Response
6.0.08	Support to Social Welfare and CIDC Support to Social Welfare and Children in Difficult Circumstances (CIDC)
6.0.99	Unallocated (Reserved for Inst.Support) Unallocated (Reserved for Institutional Support)
7	UNALLOCATED FUNDS
7.0	Unallocated Funds
7.0.00	Totally Unallocated Funds Totally Unallocated Funds

Agency Authority File

Agency	Agency Long Name	Agency Type
ACDR	Associa. Camponese Para Desenvol Ruraal	NGO
ACTION AID	Action Aid	NGO
ACTION NORD-SUD	French Action Group	NGO
ADEMO	Association For Deficiant Mozanbicans	NGO
ADOCA	Associacao Das Donas De Casa	NGO
ADPP	Ajuda Ao Povo Para O Povo	NGO
ADRA	Adventist Development & Relief Agency	NGO
AFRICARE	Africare	NGO
AICF	Action Internationale Contre La Fain	NGO
AISPO	Associa.Itali.Perla Solidar.Tra I Populi	NGO
AMAI A PABANDA	Amai A Pabanda	NGO
AMRU	Associacao Mocanbicana Da Mulher Rural	NGO
AMRU	Associacao Mocanbicana Da Mulher Rural	NGO
ARC	American Red Cross	NGO
ARO	Grupo Africa Da Suecia	NGO
ASEM	Asso..suisse Faveur L'enfante Mozanbican	NGO
ASTER	Associacao Tecnica P/Reconstruccao Nacio	NGO
ASTER	Associacao Tecnica P/Reconstruccao Nacio	NGO
BMZ	German Ministry Of Cooperation	BIL
CAPUCHINO MISSIONARY	Capuchiro Missionaries	NGO
CARE	Care International	NGO
CARITAS	Caritas	NGO
CASA DE GAIATO	Casa De Gaiato	NGO
CCM	Christian Council Of Mozambique	NGO
CEMIRDE	Commite Episcupal P/Migrantes,Refug.Desl	NGO
CHOR	Christian Outreach	NGO
CIDA	Canadian International Develop. Agency	BIL
CIES	Centro Informazione Educazione Svilupo	NGO
COCAMO	Cooperation Canada Mozambique	NGO
CONCERN	Concern	NGO
COOPI	Cooperazione Internazionale	NGO
COSV	Coordina.Organizza.Servicio Voluntario	NGO
CRIAA	Centro R./Info/Action P/Develop.en Afric	NGO
CROCEVIA	Cross Roads International Centre	NGO
CRS	Catholic Relief Services	NGO
CUAMM	Inter.College F..Health Coop.in Dev.Coun	NGO
CUSO	Canadian University Services Overseas	NGO
DHA	Department Of Humanitarian Affairs	UN
DHA/UNDRO	Department Of Humanitarian Affairs	UN
DPCCN	Dept. De Prev. E Comb...	GOM
DRC	Danish Refugee Council	NGO
DVS	Danish Voluntary Service	NGO
EMF	Eduardo Mondlane Foundation	NGO
FAR	Fellowship For African Relief	NGO
FOOD FOR THE HUNGRY	Food For The Hungry	NGO
FOS	Fonds Voor Ontwikkelings Samenwerking	NGO
GAA	German Agro Action (gaa)	NGO
GOAL	Goal	NGO
HELP AGE	Help Agency	NGO
HELVETAS	Helvetas-Switzerland	NGO
HI	Handicap Internationl	NGO
HIVOS	Hivos Foundation	NGO
IARA	Islanic African Relief Agency	NGO

Agency	Agency Long Name	Agency Type
IBIS PROJECT	Ibis Project	NGO
ICRC	International Committee Of The Red Cross	NGO
IDA	International Development Agency	NGO
IFRC	Int. Federation Of Red Cross	NGO
ISCOS	Instituto Sindical Para Cooperacao	NGO
ITALIAN COOPERATION	Italian Cooperation	BIL
ITER	Iter	NGO
JRS	Jesuit Refugee Services	NGO
KULIMA	Kulima	NGO
LRCRCS	League Of Red Cross & Red Crescent Societ	NGO
LRCS	League Red Cross/crescent Society	NGO
LWF	Lutheran World Federation	NGO
MANDRESELVA	Mandreselva	NGO
MAR'NERS	British Sea Transportation	NGO
MARINS SAN FRONTIERE	French Mariners	NGO
MBEU	Ass.Promocao Desenvol.Eco Soc.Cul.Mulher	NGO
MHC	Mozambican Health Comnity	NGO
MISSAO CONTRA LEPRA	Missao Contra A Lepra	NGO
MISSION. DE CARIDADE	Missionarios De Caridade	NGO
MOLISV	Movimento Liberazione E Sviluppo	NGO
MOZAMBICAN RED CROSS	Mozambican Red Cross	NGO
MSF(BELGIUM)	Medicine San Frontier(belgium)	NGO
MSF(FRANCE)	Medicine San Frontier(France)	NGO
MSF(HOLLAND)	Medicine San Frontier(Holland)	NGO
MSF(SPAIN)	Medicine San Frontier(Spain)	NGO
MSF(SWEDEN)	Medicine San Frontier(Sweden)	NGO
MSF(SWITZERLAND)	Medicine San Frontier(Swiss)	NGO
NCA	Norvegian Church Aid	NGO
NOVIB	Dutch Organi.for Devlop. Of Inter. Coop.	NGO
NPA	Norvegian People's Aid	NGO
NRC	Norvegian Refugee Council	NGO
OIKOS	Oikos	NGO
OXFAM	Oxfam	NGO
PROGETTO SVILUPPO	Progetto Sviluppo	NGO
PROGRESSO	Progresso	NGO
REDD BARNA	Redd Barna	NGO
RENAMO	Renamo	PAR
RRR PROJECT	Multi Churches Programme	NGO
SCF(UK)	Scf(uk)	NGO
SCF(USA)	Scf(usa)	NGO
SOS INT.	Sos International	NGO
TEARFUND	Tearfund	NGO
TERRE DES HOMMES	Lansanne	NGO
UNHCR	Un High Commission On Refugees	UN
UNICEF	Un Childrens Fund	NGO
UNILOG	Unilog	UN
VISES	Volun.Para Iniciativa,Desenvol,Econ,Soci	NGO
WFP	World Food Programme	UN
WORLD RELIEF	World Relief	NGO
WORLD VISION INT.	World Vision International	NGO
ZOA REFUGEE CARE	Z.O.A Refugee Care	NGO

NGO - Non Government Organization
GOM - Government of Mozambique
PAR - Partisans
UN - United Nations
BIL - Bilateral

Classification Authority File

ALIAS	CLASSIFY
NGO	Non-government Agency
DON	Donor
OTH	Other
MED	Media
UNM	UN Agency (Maputo)
UNH	UN Agency (Headquarters)
EMB	Embassy
GOV	Government Office
PRO	Provincial Governor
POL	Political Party

Donor Authority File

DONOR	DESCRIPTION
AUSTRALIA	Government of Australia
CANADA	Government of Canada
CHINA	Government of China
CIDA	Canadian International Development Agency
DANIDA	Danish International Development Agency
DENMARK	Government of Denmark
EEC	European Economic Community
FINLAND	Government of Finland
FINNIDA	Finland International Development Agency
FRANCE	Government of France
GERMANY	Government of Germany
ITALY	Government of Italy
ITALIAN COOPERATION	Italian Cooperation
JAPAN	Government of Japan
MOZAMBIQUE	Government of Mozambique
NETHERLANDS	Government of Netherlands
NORAD	Norwegian Agency for Development
NORWAY	Government of Norway
NRC	Norwegian Refugee Council
ODA	United Kingdom Overseas Development Agency
PORTUGAL	Government of Portugal
SDR	Swiss Disaster Relief
SIDA	Swedish International Development Agency
SOUTH AFRICA	Government of South Africa
SPAIN	Government of Spain
SPANISH COOPERATION	Spanish Cooperation
SWEDEN	Government of Sweden
SWISS COOPERATION	Swiss Cooperation
SWITZERLAND	Government of Switzerland
UNCDF	United Nations Capital Development Fund
UK	Gouernement of United Kingdom
USA	Government of United States of America
USAID	Unites States Agency for International Development

Honorific Authority File for Mailing List

HONORIFIC
Exma
Exmo
Her Excellency
His Excellency
S. Exma.

272

Incident Type Authority File

Code	Incident Type
ACCIDENT	ACCIDENT
AIR CRASH	AIR CRASH
ASSAULT	ASSAULT
HARRASSMENT	HARRASSMENT
KIDNAPPING	KIDNAPPING
LAND MINE	LAND MINE
RAPE	RAPE
SHOOTING	SHOOTING
THEFT	THEFT

Location Authority File

Location	District	Province	Latitude
Longitude	Location Type		
7 De Abril	Meconta	Nampula	15° 8 4' S (-15 14)
39° 42 6' W (39 71)	Unknown		
Agostino Neto	Homoine	Gaza	0° ' ( )
0° ' ( )	Unknown		
Alfazema - Cr13	Milange	Zambezia	16° 25 2' S (-16 42)
36° 4 8' W (36 08)	RENAMO Assembly Area		
Alto Fenfica	Mocuba	Zambezia	16° 7 8' S (-16 13)
36° 17 4' W (36 29)	Town		
Alto Changane	Chibuto	Gaza	24° 17 4' S (-24 29)
33° 37 8' W (33 63)	Town		
Alto Ligonha	Gile	Zambezia	0° ' ( )
0° ' ( )	Unknown		
Alto Molocue	Alto Molocue	Zambezia	0° ' ( )
0° ' ( )	Town		
Alua	Namapa	Nampula	13° 40 2' S (-13 67)
39° 40.2' W (39.67)	Unknown		
Amatongas	Gondola	Manica	0° ' ( )
0° ' ( )	Unknown		
Anchilo	Nampula	Nampula	0° ' ( )
0° ' ( )	Unknown		
Ancuabe	Ancuabe	Cabo Delgado	0° ' ( )
0° ' ( )	District Seat		
Angoche	Angoche	Nampula	16° 9.0' S (-16.15)
39° 32.4' W (39.54)	District Seat		
Angoche - Ng01	Angoche	Nampula	16° 13.8' S (-16.23)
39° 54.6' W (39.91)	Government Assembly Area		
Angonia	Angonia	Tete	14° 27.0' S (-14.45)
34° 9.0' W (34.15)	District Seat		
Aube	Angoche	Nampula	0° ' ( )
0° ' ( )	Unknown		
Baixo Licungo	Maganja Da Costa	Zambezia	0° ' ( )
0° ' ( )	Unknown		
Bajone	Maganja Da Costa	Zambezia	0° ' ( )
0° ' ( )	Unknown		
Balama	Balama	Cabo Delgado	0° ' ( )
0° ' ( )	Unknown		
Barue	Barue	Manica	0° ' ( )
0° ' ( )	District Seat		
Baue	Macanga	Tete	0° ' ( )
0° ' ( )	Unknown		
Beira	Dondo	Sofala	19° 30.0' S (-19.50)
34° 31.2' W (34.52)	Provincial Capital		
Bela Vista	Matutuine	Maputo	26° 12.0' S (-26.20)
32° 24.0' W (32.40)	District Seat		
Belane - Sr19	Vilanculos	Inhambane	22° 39.6' S (-22.66)
35° 15.0' W (35.25)	RENAMO Assembly Area		
Benga	Moatize	Tete	0° ' ( )
0° ' ( )	Unknown		
Bibiliza	Quissanga	Cabo Delgado	0° ' ( )
0° ' ( )	Unknown		

Location Type Authority File

Code	Location Type
AC	Airlift Center
CC	Country Capital
DS	District Seat
GAA	Government Assembly Area
PC	Provincial Capital
PRT	Port
RAA	RESAMO Assembly Area
RB	RESAMO Base
T	Town
UNA	Unattached
UNK	Unknown

Mine Type Authority File

MINE_TYPE	DESCRIPT
ANTI-TANK	Designed to explode with major pressure on it
ANTI-PERSONNEL	Designed to maim and kill individuals
ANTI-GROUP	Claymore-type mine
UXO	Unexploded ordinance
IED	Improvised explosive device
AP Ru T-46	Anti-personnel - Russian T 46

276

Mode of Transport Authority File

ALIAS	MODE
T	Truck
R	Rail
A	Air
N	Navy
B	Barge
O	Other
M	Multi-Mode
S	Ship
C	Convoy

Needs Authority File

Activity Number	Activity Title	Amount
1.0 01	Logistical Support to Repatriation	$ 11000000
2 0 01	Food	$ 4714960
2 0 01	Milling	$ 290000
2 0 01	Ocean Freight and ITSH	$ 2851554
2 0 01	Transport & Storage	$ 60294
2 0 02	Supply of Non-Food Items & Accomodations	$ 3161500
2 0 03	Equipment	$ 500000
2.0.03	General Operational Expenses	$ 300000
2.0.03	Medical Spplies	$ 1000000
2.0.03	Miscellaneous	$ 476022
2.0.03	Salaries	$ 1700000
2.0.04	Implementation of works	$ 282637
2 0 04	Purchase supplies	$ 219829
2 0 04	Technical assistance	$ 251234
2.0.05	Civilian Clothing	$ 2513430
2.0.05	Transport	$ 208257
2.0.06	Registration & Database	$ 300000
2.0.07	General administration	$ 300000
2.0.07	Office and equipment	$ 1016000
2.0.07	Operational costs	$ 11962210
2 0 07	Personnel costs	$ 1810000
2.0.08	Demobilization subsidies	$ 22470000
2.0.09	Technical Unit	$ 1312747
2.0.10	Literacy & Printed Materials	$ 143500
2 0.10	Personnel	$ 84800
2.0.10	Radio	$ 107000
2.0.10	Theatre, Cultural & Sports Activities	$ 41500
2.0 11	Family Reunification	$ 860000
2.0.11	Logistics and Monitoring	$ 97000
2.0 11	Personnel	$ 174000
2 0 11	Registration , Information for pension disabled	$ 119000
2.0.11	Support for Reintegration	$ 275000
3 1.01	Provision of cereals	$ 29704700
3.1.01	Provision of non-cereals	$ 21123200
3 1 02	Food for work programmes	$

Activity Number	Activity Title	Amount
3.1.03	Commodities for hospital diet	$ 2500000
3.1.03	Commodities for therapeutic feeding	$ 7190000
3.2.01	Road, rail and sea transport	$ 41576769
3.2.02	Airlift operations	$ 2500000
3.2.03	Fuel and lubricants	$ 768000
3.2.03	Spares	$ 600000
3.2.03	Storage	$ 2000000
3.3.01	Office costs	$ 1024000
3.3.01	Operational costs	$ 5225000
3.3.01	Staff	$ 1145168
3.4.01	Logistics and storage	$ 1146750
3.4.01	Purchase of relief and survival items	$ 4587000
3.4.01	Purchase of water jars, tents and plastic sheeting	$ 2000000
3.4.02	Logistics and distribution	$ 2077500
3.4.02	New equipped warehouses	$ 595000
3.4.02	Purchase of relief items	$ 8310000
3.4.02	Technical assistance and operation costs	$ 828800
4.1.02	Logistic and administrative support	$ 500000
4.1.02	Promotion of local seed multiplication	$ 1000000
4.1.02	Purchase of seeds	$ 37114000
4.1.02	Seed quality control	$ 100000
4.1.02	Transport and storage	$ 3925000
4.1.03	Purchase of tools	$ 47831314
4.1.03	Transport and storage	$ 2000000
4.1.04	Pest control	$ 1100000
4.1.04	Seed inspection	$ 250000
4.1.05	Support to INIA's and district centers	$ 1100000
4.1.05	Transportation and supervision	$ 400000
4.1.06	Local purchase (including ITSH)	$ 12772000
4.1.06	Operation funds (at average USD 20,000/province)	$ 200000
4.1.06	Training and orientation	$ 50000
4.1.07	Local impact projects	$ 350000
4.1.07	Recruitment and transportation of extension agents	$ 1000000
4.1.07	Training and logistics	$ 150000
4.1.08	Contractual services	$ 1298000
4.1.08	Operational expenses	$ 15000
4.2.01	Drugs for response to epidemics	$ 400000
4.2.01	Drugs for rural hospitals; in-patient level	$ 2000000
4.2.01	EDP and WHO emergency for PHC level	$ 1600000
4.2.01	Support to MEDIMOC's imports	$ 5000000
4.2.02	Cold-chain equipment and other supplies	$ 800000
4.2.02	Vaccines and consumables	$ 800000
4.2.03	Equipment: transport, communication, office	$ 400000
4.2.03	General operating expenses	$ 600000
4.2.03	Local costs	$ 800000
4.2.03	Technical assistance	$ 1000000

Activity Number	Activity Title	Amount
4.2 03	Training and travel expenses	$ 700000
4.2.04	General operating expenses	$ 1200000
4.2.04	Hospital equipment and furniture	$ 750000
4.2.04	Medical supplies and sterilization equipment	$ 3250000
4.2.04	Rehabilitation of health facilities	$ 6100000
4.2.04	Technical assistance and NGO contracts	$ 2000000
4.2.05	Contract of NGOs Services: Salaries, commod.,etc.	$ 18000000
4.2.05	Support to provincial health authorities	$ 500000
4.2.06	Material and equipment for workshops	$ 1300000
4.2.06	NGOs contracts and technical assistance	$ 2500000
4.2.06	Operating expenses, duty travel for supervision	$ 800000
4.2.06	Rehabilitation works	$ 800000
4.2.06	Training activities	$ 350000
4.2.06	Vehicles, communications, office equip., logistics	$ 250000
4.3.01	AFRIDEV handpumps	$ 1224000
4.3.01	Percusion drilling rigs & hand drilling rigs	$ 5660000
4.3.01	Sanitation	$ 1250000
4.3.01	Vehicles and spares (large/small trucks - 70)	$ 3225000
4.3.01	Well and borehole const.equipment and costs	$ 3066000
4.4.01	Construction - DNEP and NGOs	$ 20000000
4.4.01	Survey, design and supervision	$ 14600000
4.5.01	Functional rehabilitation of 1,000 classrooms	$ 5000000
4.5.01	Provision of 500 teachers kits	$ 75000
4.5.01	Provision of material and equipment kits	$ 3500000
4.5.01	Purchase school text books	$ 1000000
4.6.01	Area-based programmes at community level	$ 25000000
4.6.02	QUIPS	$ 13987000
4.6.03	CORE's Programme Development	$ 380000
4.6.03	Information/Counselling Services	$ 1600000
4.6.03	Veterans Association(s)	$ 500000
4.6.03	Vocational Training Instit.	$ 2583000
4.6.04	Vocational/Technical/Entrepreneurial/Training	$ 6484000
4.6.05	Follow-up activity loans	$ 3500000
4.6.05	Group/association loans	$ 2000000
4.6.05	Kits and Credit Admin Costs	$ 500000
4.6.05	Small-business loans	$ 4000000
4.6.05	Vocational/Activity kits	$ 10000000
4.6.06	Job Identification and Support	$ 1400000
4.6.06	Labour-Intensive Work District Level Fund	$ 1000000
4.6.06	Vegetable Kits	$ 765250
4.7.01	Cost of survey contract	$ 420000
4.7.01	Mine clearance contracts	$ 25000000
4.7.01	Monitoring and quality assurance	$ 1400000
4.7.01	Salary and equipment costs	$ 482000
4.7.02	Building and miscelaneous costs	$ 655640
4.7.02	Equipment costs	$ 953895
4.7.02	Personnel Costs	$ 2206000
4.7.02	UNHCR mine awareness campaign	$ 500000

Activity Number	Activity Title	Amount
5.0.01	Maize	$ 22000000
5.0.01	Rice	$ 18956000
5.0.01	Wheat	$ 18102000
6.0.01	Improvement of communication network	$ 95000
6.0.01	Provincial food security data collection network	$ 70000
6.0.01	Rehabilitation of meteorological stations	$ 500000
6.0.02	Support to CENE/CPEs and CDEs	$ 2000000
6.0.03	Administration and finance	$ 800000
6.0.03	Planning and information	$ 700000
6.0.03	Support to provincial DPCCN	$ 600000
6.0.03	Technical assistance to LSU	$ 3000000
6.0.04	Central task force, duty travel and local costs	$ 625000
6.0.04	Equipment, transport, communications and offices	$ 30000
6.0.04	General expenses, office costs, documentation	$ 100000
6.0.04	International recruitment	$ 170000
6.0.04	Local contracts	$ 30000
6.0.04	Short term consultancies	$ 40000
6.0.05	NGO fund	$ 300000
6.0.06	Financial assistance to NAR	$ 4000000
6.0.07	Provision for Technical Assistance	$ 300000
6.0.07	Training Seminars upgrading 60 teachers	$ 18000
6.0.07	Transport and Logistics	$ 295000
6.0.08	Advocacy and social communication	$ 300000
6.0.08	Institutional support	$ 300000
6.0.08	Local recruitment of central task force & costs	$ 300000
6.0.08	Re-integration of vulnerable groups and CIDC	$ 1220000
6.0.08	Rehabilitation of social action facilities	$ 1200000
6.0.08	Support to NGO network	$ 900000
6.0.08	Technical assistance	$ 130000

Resource Type Authority File

Type	Sector
SCHOOL	EDUCATION
HEALTH POST/CENTER	HEALTH
HOSPITAL	HEALTH
SHOPS & STORES	MARKETING
WELLS & BOREHOLES	WATER

Title Authority File for Mailing List

Title
Dr.
Mr.
Mrs.
Ms
Sr.
Sra.

Activity Number	Title	Total Needs US$	Commitments US$	Balance US$	Obligated to Disburse US$
5	BALANCE OF PAYMENTS AND BUDGET SUPPORT				
	Balance of Payment Support				
5.0.01	Market Food Aid	59,058,000	41,992,869	17,065,131	32,472,869
5.0.02	Balance of Payments - Seeds & Tools		1,500,000	(1,500,000)	1,500,000
	Sector Totals	59,058,000	43,492,869	15,565,131	33,972,869
Category Totals		59,058,000	43,492,869	15,565,131	33,972,869
6	INSTITUTIONAL SUPPORT				
	Institutional Support				
6.0.00	Allocated to General Institut. Support		967,750	(967,750)	0
6.0.01	Food Security & Early Warning Systems	665,000	14,139	650,861	14,139
6.0.02	Support to CPEs and CDEs	2,000,000	1,110,320	889,680	1,110,320
6.0.03	Institutional Support to DPCCN	5,100,000	2,502,674	2,597,326	2,502,674
6.0.04	MOH Emergency Response	995,000	99,609	895,391	99,609
6.0.05	National NGOs	300,000	718,075	(418,075)	438,075
6.0.06	Support to NAR	4,000,000	3,271,948	728,052	1,870,000
6.0.07	Ministry of Education Emergency Response	613,000		613,000	0
6.0.08	Support to Social Welfare and CIDC	4,350,000	563,308	3,786,692	357,300
	Sector Totals	18,023,000	9,247,823	8,775,177	6,392,117
Category Totals		18,023,000	9,247,823	8,775,177	6,392,117
7	UNALLOCATED FUNDS				
	Totally Unallocated Monies				
7.0.00	Totally Unallocated Funds		8,247,586	(8,247,586)	7,414,286
	Sector Totals		8,247,586	(8,247,586)	7,414,286
Category Totals			8,247,586	(8,247,586)	7,414,286

SUMMARY TOTALS	Needs	Total Commitments	Balance	Obligated to Disburse
	609,688,760	549,084,801	60,603,959	360,946,515

Notes: 1 - **Allocated to a General Project (x.x.00):** Commitments to a a project or programme cannot be fitted into an activity in the Consolidated Programme.

2 - **Unallocated (Reserved for Sector/Category) (x.x.99):** Committments to the sector, pending identification of the specific activity.

3 - **Totally Unallocated Funds (7.0.00):** Committed to Consolidated Humanitarian Programme in general, pending identification of the sector and specific activity.

4 - **Grand Total Unallocated (x.x.99 + 7.0.00) = $31,241,905**

Activity Number	Title	Total Needs US$	Commitments US$	Balance US$	Obligated to Disburse US$
4	RESTORATION OF ESSENTIAL SERVICES				
	Restoration of Services				
4.0.00	Allocated to Restoration Services		25,840,500	(25,840,500)	4,062,000
	Sector Totals		25,840,500	(25,840,500)	4,062,000
	Agriculture				
4.1.01	Seeds & Tools (see 4.1.02 & 4.1.03)		5,678,432	(5,678,432)	3,017,812
4.1.02	Seed for Family Sector	42,639,000	38,481,490	4,157,510	36,808,967
4.1.03	Supply of Agricultural Tools	49,831,314	18,076,418	31,754,896	18,038,918
4.1.04	Pest Control and Seed Inspection	1,350,000		1,350,000	0
4.1.05	Cassava Production	1,500,000	14,345	1,485,655	14,345
4.1.06	Marketing of Agricultural Produce	13,022,000	4,201,082	8,820,918	3,989,060
4.1.07	Crash Agricultural Extension Programme	1,500,000	457,772	1,042,228	86,072
4.1.08	Reserve Grain Silo, Beira	1,343,000		1,343,000	0
4.1.99	Unallocated (Reserved for Agriculture)		140,000	(140,000)	140,000
	Sector Totals	111,185,314	67,049,599	44,135,715	62,095,174
	Health				
4.2.01	Stockpiles of Drugs/Reagents	9,000,000	3,708,667	5,291,333	1,668,787
4.2.02	Cold Chain Equipment/Vaccines	1,600,000	1,549,642	50,358	1,549,642
4.2.03	Support to Provinc. Health Directorates	3,500,000	1,699,866	1,800,134	1,628,589
4.2.04	Rehabilitation of Health Network	13,300,000	2,988,391	10,311,609	1,278,811
4.2.05	Rural Health Care	18,500,000	24,467,046	(5,967,046)	23,733,916
4.2.06	Prosthetics Assistance	6,000,000	7,930,000	(1,930,000)	5,900,000
	Sector Totals	51,900,000	42,343,612	9,556,388	35,759,745
	Water Supply and Sanitation				
4.3.01	Water Supply for Resettlement	14,425,000	10,061,930	4,363,070	9,149,930
	Sector Totals	14,425,000	10,061,930	4,363,070	9,149,930
	Roads and Bridges				
4.4.01	Roads and Bridges	34,600,000	48,948,622	(14,348,622)	29,727,728
	Sector Totals	34,600,000	48,948,622	(14,348,622)	29,727,728
	Education				
4.5.01	Education and Teachers Training	9,575,000	3,912,011	5,662,989	3,276,000
	Sector Totals	9,575,000	3,912,011	5,662,989	3,276,000
	Multi-Sectoral Programmes				
4.6.00	Alloc. to a General Multi-Sectoral Prog.		2,554,600	(2,554,600)	550,000
4.6.01	Area Based Programmes at Community Level	25,000,000	24,968,514	31,486	5,697,714
4.6.02	Support to Returnee Areas (QIPS)	13,987,000	8,571,300	5,415,700	6,387,000
4.6.03	Institutional Support for Demob Soldiers	5,063,000	3,352,800	1,710,200	3,352,800
4.6.04	Training for Demobilized Soldiers	6,484,000	2,700,000	3,784,000	1,500,000
4.6.05	Kits and Credit Scheme for Demobilized S	20,000,000	1,200,000	18,800,000	0
4.6.06	Employment for Demobilized Soldiers	3,165,250	2,172,250	993,000	0
4.6.99	Unallocated (Reserved for Multi-Sector)		4,500,000	(4,500,000)	0
	Sector Totals	73,699,250	50,019,464	23,679,786	17,487,514
	Mine Clearance				
4.7.01	Mine Survey & Road Mine Clearance	27,302,000	24,341,411	2,960,589	23,584,411
4.7.02	Mine Awareness & Clearance Training	4,315,535	3,192,035	1,123,500	3,192,035
4.7.99	Unallocated (Reserved for Mine Clearance		605,965	(605,965)	0
	Sector Totals	31,617,535	28,139,411	3,478,124	26,776,446
Category Totals		327,002,099	276,315,149	50,686,950	188,334,537

Activity Number	Title	Total Needs US$	Commitments US$	Balance US$	Obligated to Disburse US$
1	REPATRIATION OF REFUGEES				
	Repatriation Operation				
1.0.01	Repatriation Operation	11,000,000	5,400,000	5,600,000	5,400,000
	Sector Totals	11,000,000	5,400,000	5,600,000	5,400,000
Category Totals		11,000,000	5,400,000	5,600,000	5,400,000
2	DEMOBILIZATION				
	Demobilization				
2.0.01	Supply of Food For Demobilization	7,916,808	7,916,808	0	7,916,808
2.0.02	Supply of Non-food Items for Demobilizat	3,161,500	3,161,500	0	1,415,438
2.0.03	Health Care for Demobilization	3,976,022	3,946,022	30,000	3,056,168
2.0.04	Water Supply and Sanitation for Demobil.	753,700	1,421,082	(667,382)	975,700
2.0.05	Civilian Clothing for Demobilization	2,721,687	2,721,687	0	1,433,848
2.0.06	Registration of Demobilizing Soldiers	300,000	300,000	0	17,384
2.0.07	Home Transportation for Demobilized	15,088,210	13,285,661	1,802,549	8,285,661
2.0.08	Demobilization Subsidies	22,470,000	22,470,000	0	14,600,000
2.0.09	Technical Unit for Demobilization	1,312,747	1,312,747	0	1,165,000
2.0.10	Information & Social Reintegrat.	477,800	477,800	0	89,604
2.0.11	Demob & Reintegrat. of Vulnerable Groups	1,525,000	1,525,000	0	665,000
2.0.99	Unallocated (Reserved for Reintegration		5,907,640	(5,907,640)	0
	Sector Totals	59,703,474	64,445,947	(4,742,473)	39,620,611
Category Totals		59,703,474	64,445,947	(4,742,473)	39,620,611
3	EMERGENCY RELIEF				
	Emergency Relief				
3.0.00	Allocated to General Project-Emergency		3,681,286	(3,681,286)	3,404,836
3.0.99	Unallocated (Reserved for Emerg.Relief)		285,714	(285,714)	285,714
	Sector Totals		3,967,000	(3,967,000)	3,690,550
	Relief Food Aid				
3.1.00	Allocated to Relief Food		6,337,650	(6,337,650)	1,322,650
3.1.01	Food Relief Distributions	50,827,900	53,719,671	(2,891,771)	33,452,171
3.1.02	Food for Work		2,673,152	(2,673,152)	665,652
3.1.03	Nutritional Rehabilitation	9,690,000	14,733,769	(5,043,769)	9,356,469
3.1.99	Unallocated (Reserved for Relief Food)		10,250,000	(10,250,000)	2,250,000
	Sector Totals	60,517,900	87,714,242	(27,196,342)	47,046,942
	Logistics				
3.2.00	Allocated to a General Logistics Project		743,472	(743,472)	502,306
3.2.01	Road, Rail and Sea Transport	41,576,769	27,602,459	13,974,310	18,475,714
3.2.02	Airlift Operations	2,500,000	3,833,883	(1,333,883)	833,883
3.2.03	Maintenance and Storage	3,368,000	5,222,134	(1,854,134)	5,222,134
3.2.99	Unallocated (Reserved for Logistics)		905,000	(905,000)	0
	Sector Totals	47,444,769	38,306,948	9,137,821	25,034,037
	Transport of Vulnerable Groups				
3.3.01	Transport of Vulnerable Groups	7,394,468	2,107,692	5,286,776	707,692
	Sector Totals	7,394,468	2,107,692	5,286,776	707,692
	Non-Food Relief				
3.4.00	Allocated to General Non-Food Relief		2,600,000	(2,600,000)	0
3.4.01	Returnees and Internally Displaced	7,733,750	3,775,323	3,958,427	1,748,652
3.4.02	Emergency Stockpile Project	11,811,300	3,064,222	8,747,078	1,264,222
3.4.99	Unallocated (Reserved for Non-Food)		400,000	(400,000)	320,000
	Sector Totals	19,545,050	9,839,545	9,705,505	3,332,874
Category Totals		134,902,187	141,935,427	(7,033,240)	79,812,095

United Nations Office for Humanitarian Assistance Coordination

Committments to Humanitarian Assistance Programme
May 1993 - April 1994

FINANCIAL SUMMARY

SUMMARY TOTALS	Needs	Total Commitments	Balance	Obligated to Disburse
	609,688,760	549,084,801	60,603,959	360,946,515

United Nations Office for Humanitarian Assistance Coordination

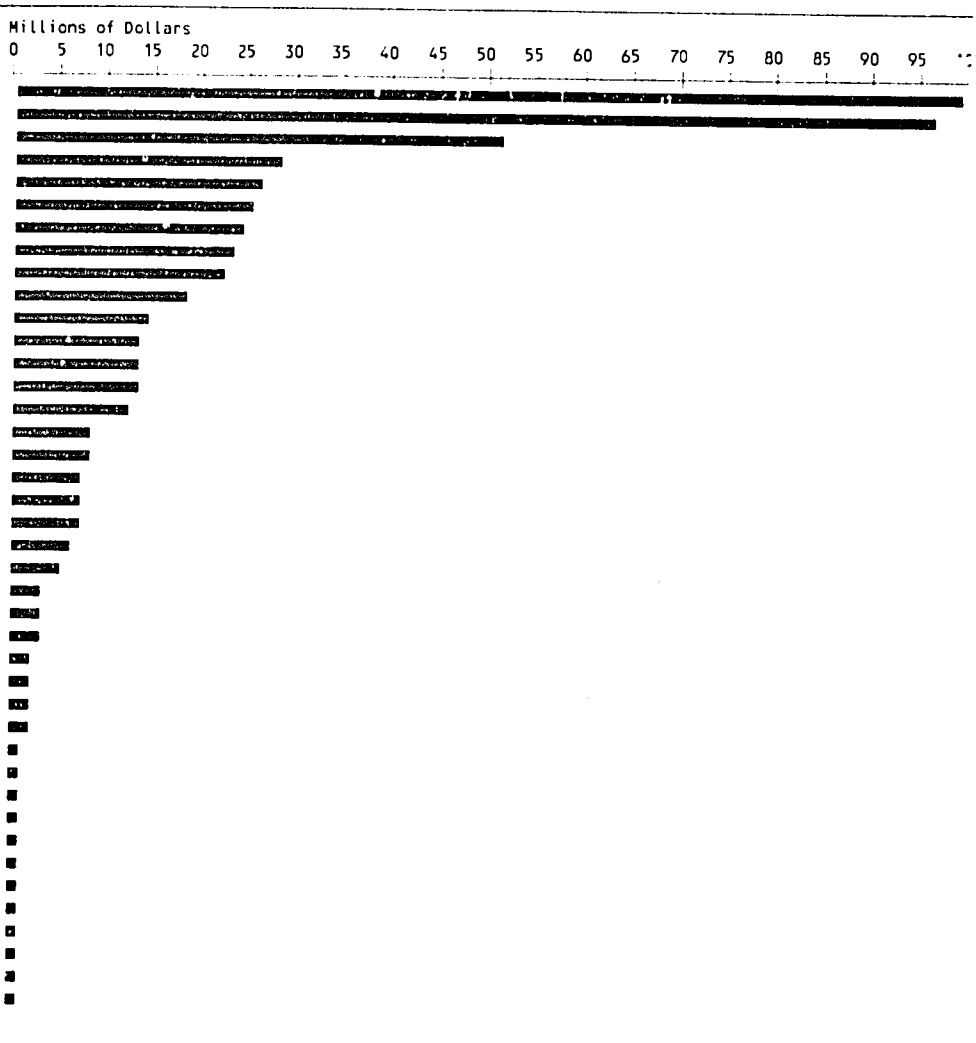
SHAMAN: System for Humanitarian Assistance Management

CONTRIBUTIONS BY USD AMOUNT

Current to: 28 December 1993

Ranked by Donor by Amount

OHAC SHAMAN — CONTRIBUTIONS BY $ AMOUNT — 28 December 1993



EEC	99,178,130
USA	96,481,096
WFP	50,693,757
ITALY	27,949,844
NORWAY	25,472,591
NETHERLANDS	24,892,678
GERMANY	23,521,786
SWEDEN	22,525,923
MOZAMBIQUE	22,470,000
CANADA	17,968,300
UNHCR	14,493,300
UK	13,323,888
JAPAN	13,240,000
DENMARK	13,067,396
UNUMOZ	11,967,965
SWITZERLAND	8,191,500
EDA	7,500,000
SOUTH AFRICA	7,423,897
UN	7,000,000
UNICEF	6,815,251
SPAIN	6,063,400
FRANCE	5,480,000
UNOMOZ	3,032,035
PORTUGAL	3,002,000
ITI	2,900,000
CEA	2,000,000
AUSTRALIA	1,792,000
IRF	1,640,900
IRAID	1,500,000
UNA	1,474,543
OXF(UK)	1,285,454
WORLD VISION	963,600
UNA	792,000
UNDP	437,000
FINLAND	409,000
UNA	400,000
	200,000
CARE(FRANCE)	195,000
OXFAM	169,286
	137,833
	33,448

Total Carryover 92/93: 50,119,619

Total Commitments 93/94: 498,965,182

Grand Total: 549,084,801

Appendix D

Sample Reports

Contributions by USD Amount
Consolidated Humanitarian Assistance Programme
Activity Summaries
Constributions by Activity
Contributions by Donor
Financial Summary
Needs Summary
Mine Assessment and Survey
Population 1980,Dec '92, Dec '93
Relief Dispatches to RENAMO Areas by Agency
Relief Dispatches to RENAMO Areas by Province
Who is Where
Who-What-Where by Province, District, Agency
and Activity
Who is Doing What Where by Province, District,
Agency and Activity (R&R Windows Report)

UNOHAC Who's Doing What Where Report Form

United Nations Office for Humanitarian Assistance Coordination

9th Floor, Rovuma Hotel, Maputo - Fax 42 51 36

Tel: 42 32 17, 42 33 94/6/9, 42 99 13/4./5/6, 42 99 34/5/6/7/8/9, 42 99 43/4 Ext 2233

Agency: _____ Phone: _____ No. of Staff: _____ Date: _____

Address: _____ Fax: _____ Fiscal Year Start: _____ (dd/mm/yy)

City: _____ Province: _____ Contact: _____

District: _____ Contact Title: _____

Please chose Activity No. and Activity Name from List on Back

Activity No.	Activity Name	Specific Place	District(s)	Starting Date	Ending Date	Brief Description of Activity

Remarks:

Remarks:

Remarks:

Remarks:

SEE LIST ON REVERSE TO CHOOSE ACTIVITY NO. AND ACTIVITY NAME

1.0. REPATRIATION OF REFUGEES

1.0.00 Repatriation Operation

2.0. DEMOBILIZATION

2.0.00 General Allocated - Demobilization
2.0.01 Supply of Food for Demobilization
2.0.02 Supply on Non-food Items
2.0.03 Health Care
2.0.04 Water Supply and Sanitation
2.0.05 Civilian Clothing
2.0.06 Registration
2.0.07 Home Transportation
2.0.08 Demobilization Subsidies
2.0.09 Technical Unit
2.0.10 Organized Activities Entertainment
2.0.99 Unallocated - Demobilization

3.0. EMERGENCY RELIEF

3.0.00 General Allocated - Emergency Relief
3.0.99 Unallocated- Emergency Relief

3.1. Relief Food AID

3.1.00 General Allocated - Relief Food Aid
3.1.01 Food Relief Distributions
3.1.02 Food for Work
3.1.03 Nutritional Rehabilitation
3.1.99 Unallocated- Relief Food Aid

3.2. General Allocated - Food Aid Logistics

3.2.01 Road, Rail and Sea Transport
3.2.02 Airlift Operations
3.2.03 Maintenance and Storage
3.2.99 Unallocated - Logistics

3.3. Transport of People

3.3.00 General Allocated - Transport of People
3.3.01 Transport of Vulnerable Groups

3.4. Non-Food Relief

3.4.00 General Allocated - Non Food Relief
3.4.01 Returnees and Internally Displaced
3.4.02 Emergency Stockpile Project
3.4.99 Unallocated - Non-Food Relief

4.0. RESTORATION OF ESSENTIAL SERVICES

4.0.00 General Allocated - Restoration of Service
4.0.99 Unallocated - Restoration of Services

4.1 Agriculture

4.1.00 General Allocated - Agriculture
4.1.01 Seeds & Tools *(please divide below)*
4.1.02 Seed for Family Sector
4.1.03 Agricultural Tools
4.1.04 Pest Control and Seed Inspection
4.1.05 Cassava Production
4.1.06 Marketing of Agricultural Produce
4.1.07 Crash Extension Programmme
4.1.08 Reserve Grain Silo, Beira
4.1.99 Unallocated - Agricultural

4.2. Health

4.2.00 General Allocated - Health
4.2.01 Stockpiles of Drugs/Reagents
4.2.02 Cold Chain Equipment/Vaccines
4.2.03 Support to Provincial Directorates
4.2.04 Rehabilitation of Health Network
4.2.05 Rural Health Care
4.2.06 Prosthetics Assistance
4.2.99 Unallocated Health

4.3. Water

4.3.01 Water Supply for Resettlement

4.4. Roads and Bridges

4.4.01 Roads and Bridges

4.5. Education

4.5.00 General Allocated Education
4.5.01 Education and Teacher Training

4.6. Multisectoral Programs

4.6.00 General Allocated - Multisectoral Programme
4.6.01 Area Based Programmes at Community Level
4.6.02 Support to Returnee Areas (QIPs)
4.6.03 Reintegration Programme for Demobilized Soldiers
4.6.04 Support to Social Welfare and CIDC
4.6.99 Unallocated - Multisectoral Programs

4.7. Mine Clearance

4.7.00 General Allocated - Mine Clearance
4.7.01 Mine Survey and Road Mine Clearance
4.7.02 Mine Awareness and Mine Clearance
4.7.99 Unallocated - Mine Clearance

5.0. BALANCE OF PAYMENTS AND BUDGET SUPPORT

5.0.00 General Allocated - Balance of Payments
5.0.01 Market Food Aid
5.0.02 Balance of Payment - Seeds and Tools
5.0.99 Unallocated - Balance of Payment Support

6.0. INSTITUTIONAL SUPPORT

6.0.00 General Allocated - Institutional Support
6.0.01 Food Security & Early Warning Systems
6.0.02 Support to CPEs & CDEs
6.0.03 Institutional Support to DPCCN
6.0.04 MOH Emergency Response
6.0.05 National NGOs
6.0.06 Support to NAR
6.0.09 Unallocated- Institutional Support

7.0. Unallocated Funds

7.0.00 Unallocated Funds

Contribution Report Form

United Nations Office for Humanitarian Assistance Coordination
9th Floor, Rovuma Hotel, Maputo - Fax 42 51 36
Tel 42 32 17 42 33 94/6/9, 42 99 13/4 /5/6, 42 99 34/5/6/7/8/9, 42 99 43/4 Ext 2218

Report Date: ____/____/____ (dd/mm/yy) *(For UNOHAC internal use)* UNOHAC Reference No: _____

DONOR DETAILS

Donor Organization: _____ Donor Project No: _____

Donor Country: _____ Contact Person(s): _____

Donor Fiscal Year Starts: ____/____/____ Telephone: _____

Implementing Agency: _____

○ Gov't ○ UN Agency ○ Bilateral
○ NGO ○ Undecided *(Check one)*

ACTIVITY DESCRIPTION

Programme Activity No: _____ Programme Activity Title: _____
(See reverse)

Remarks: *(A few lines to help the reader understand the commitment information)*

Activity Start Date : ____/____/____ End Date: ____/____/____ (dd/mm/yy)

CONTRIBUTION AMOUNT AND DETAILS

Your efforts to estimate geographical breakdown of amount, especially by district, will be greatly appreciated. Thank you.

Province	District(s)	Quantity *(if applicable)*	Unit (Tons, etc)	US$ Amount
TOTAL(s)				(Total Commitment)

CONTRIBUTION STATUS 1993/94

Choose only one
(Divide into more Reports if needed)

○ Interest Expressed
or
Commitment 93/94 Is this

(Firm Programme intention but not necessarily a signed Project document)

○ 1992/93 Carryover Funds
and/or
○ 1993/94 Commitment
(Estimate if necessary)

If both, give amounts US$ _____
(Sum equals Total Commitment)

Disbursements or Obligations for Disbursements ○ 100% or _____
of Total Commitment

Appendix C

Authority Tables

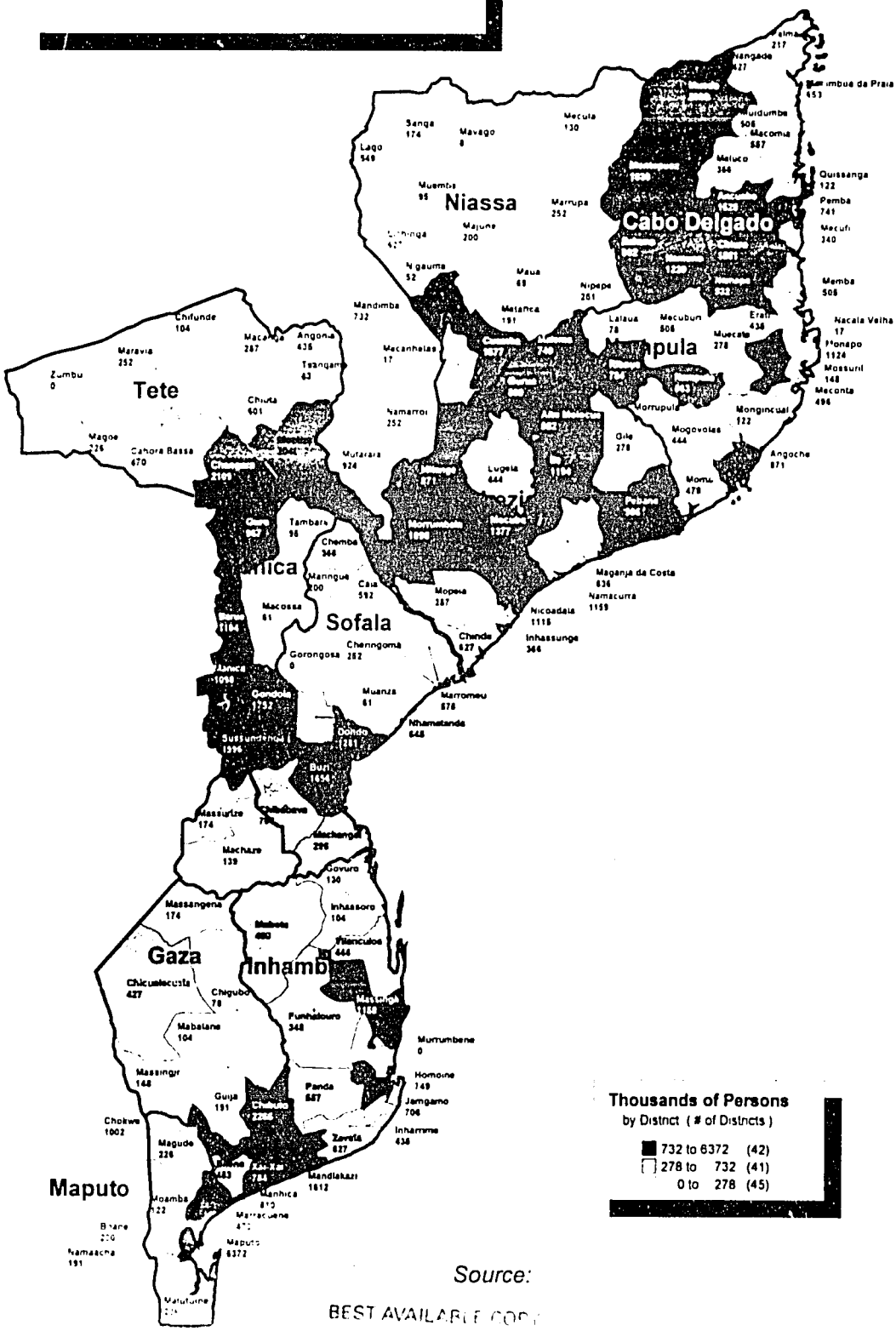
Activity Title
Agency
Classification
Donor
Honorific
Incident Type
Location
Location Type
Mine Type
Mode of Transport
Needs
Resource Type
Title

Appendix F

Sample Forms

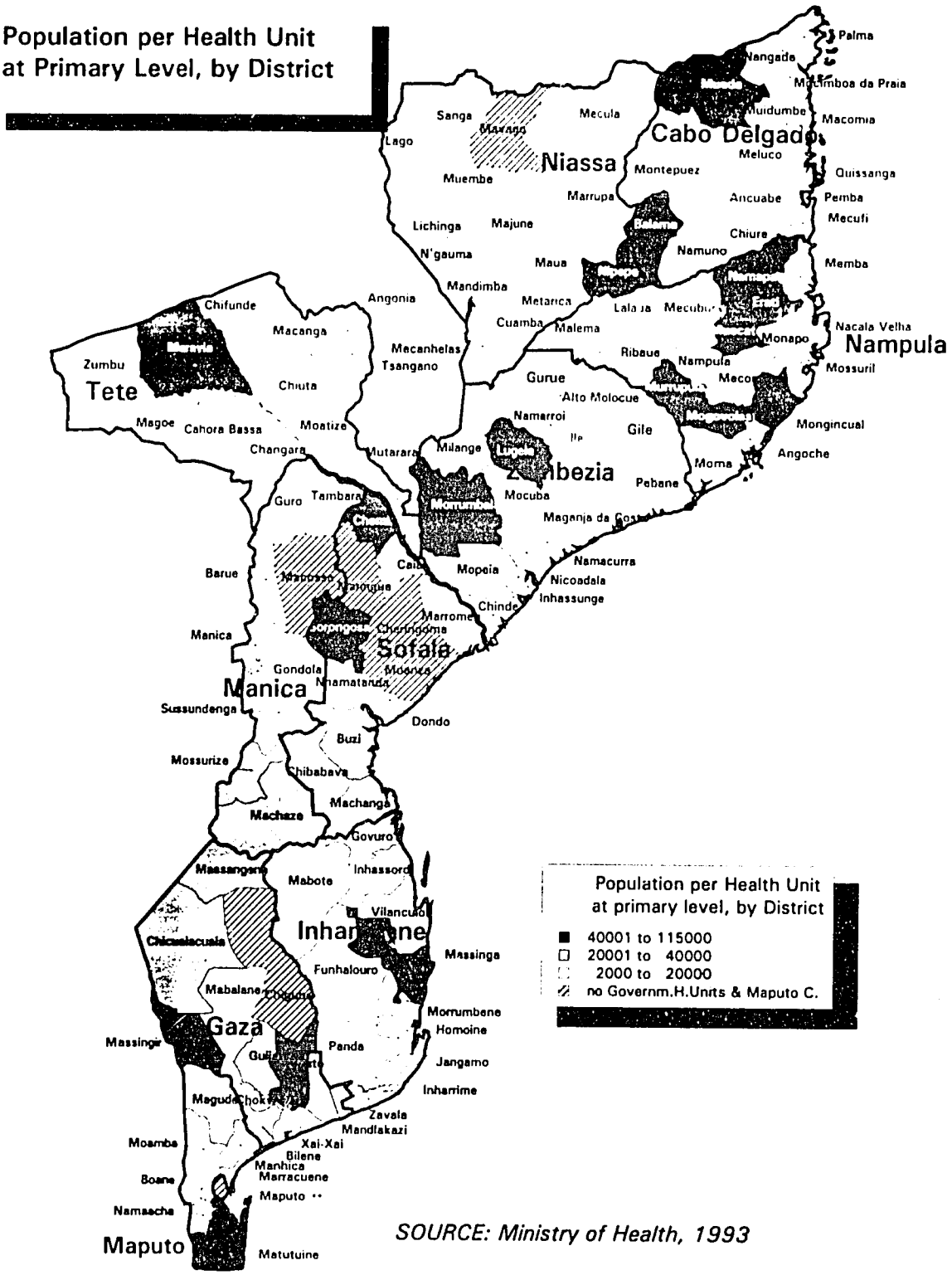
Republic of Mozambique

Projected Destination of 85,000 Demobilizing Soldiers



Source:

Population per Health Unit at Primary Level, by District



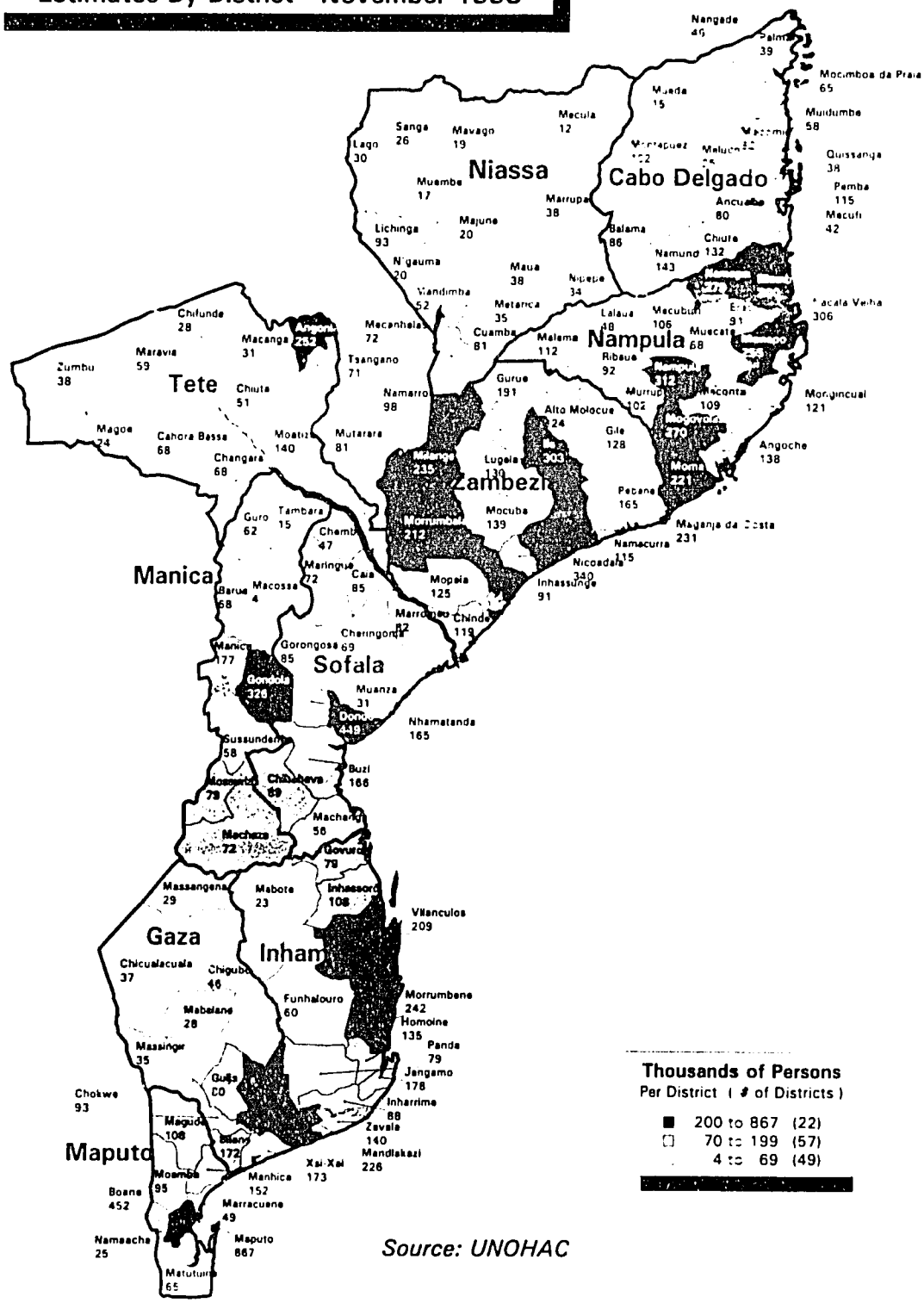
SOURCE: Ministry of Health, 1993

** *no primary health facilities were included for Maputo City*

United Nations Office for Humanitarian Assistance Coordination

December 1993

Population Inside Mozambique
Estimates By District - November 1993

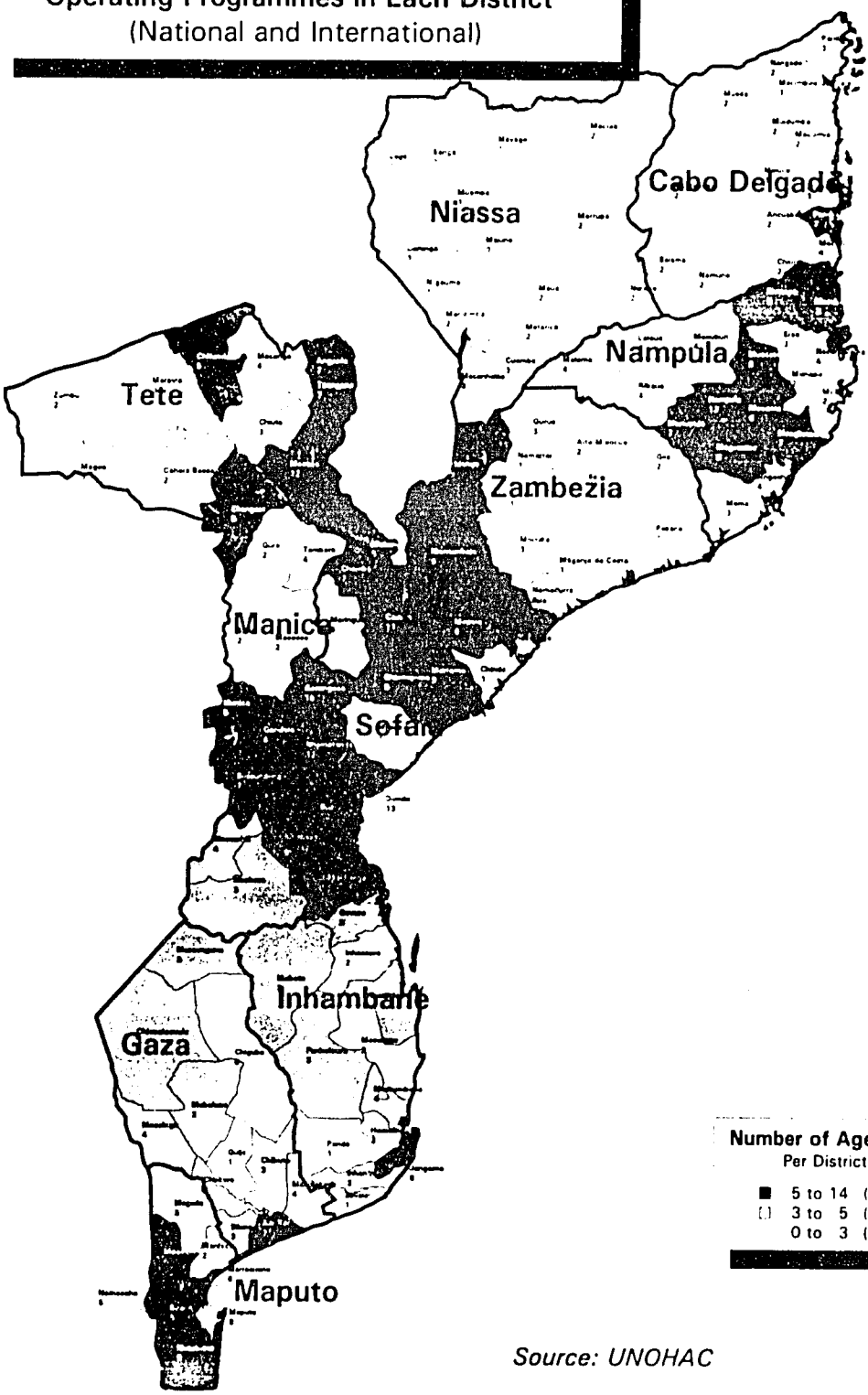


Source: UNOHAC

United Nations Office for Humanitarian Assistance Coordination

December 1993

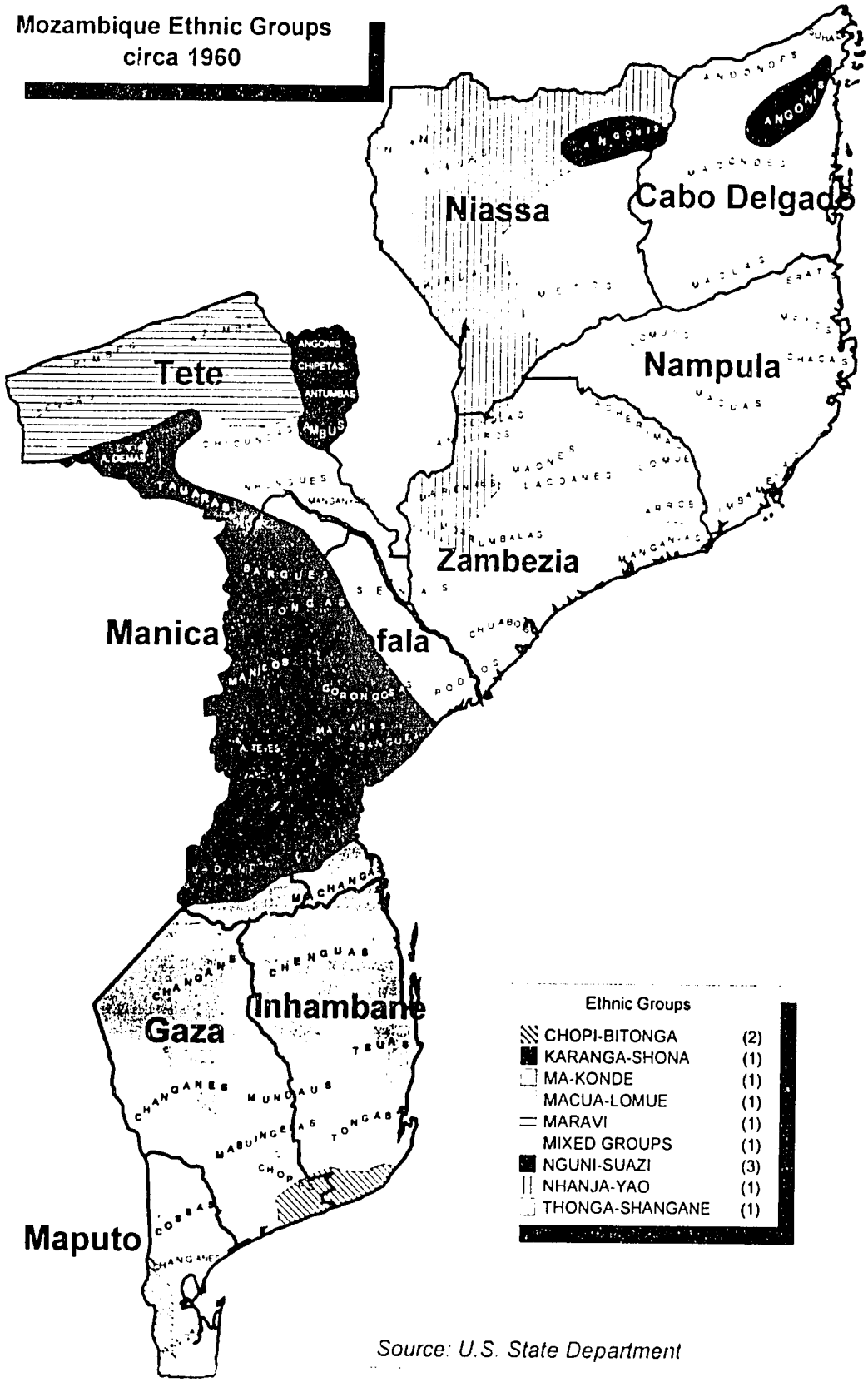
Number of NGOS's and Other Agencies Operating Programmes in Each District
(National and International)



Source: UNOHAC

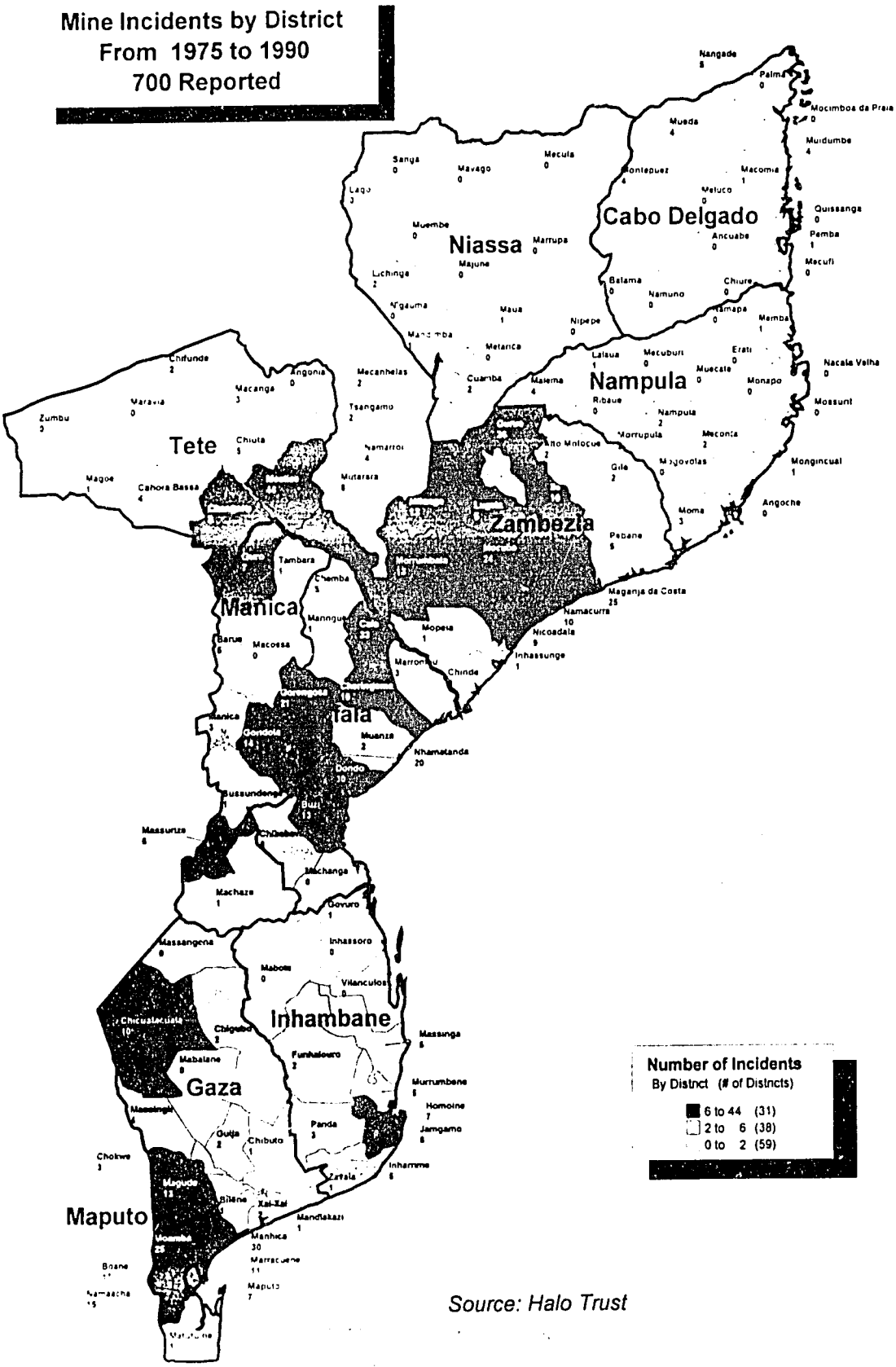
United Nations Office for Humanitarian Assistance Coordination

November 1993

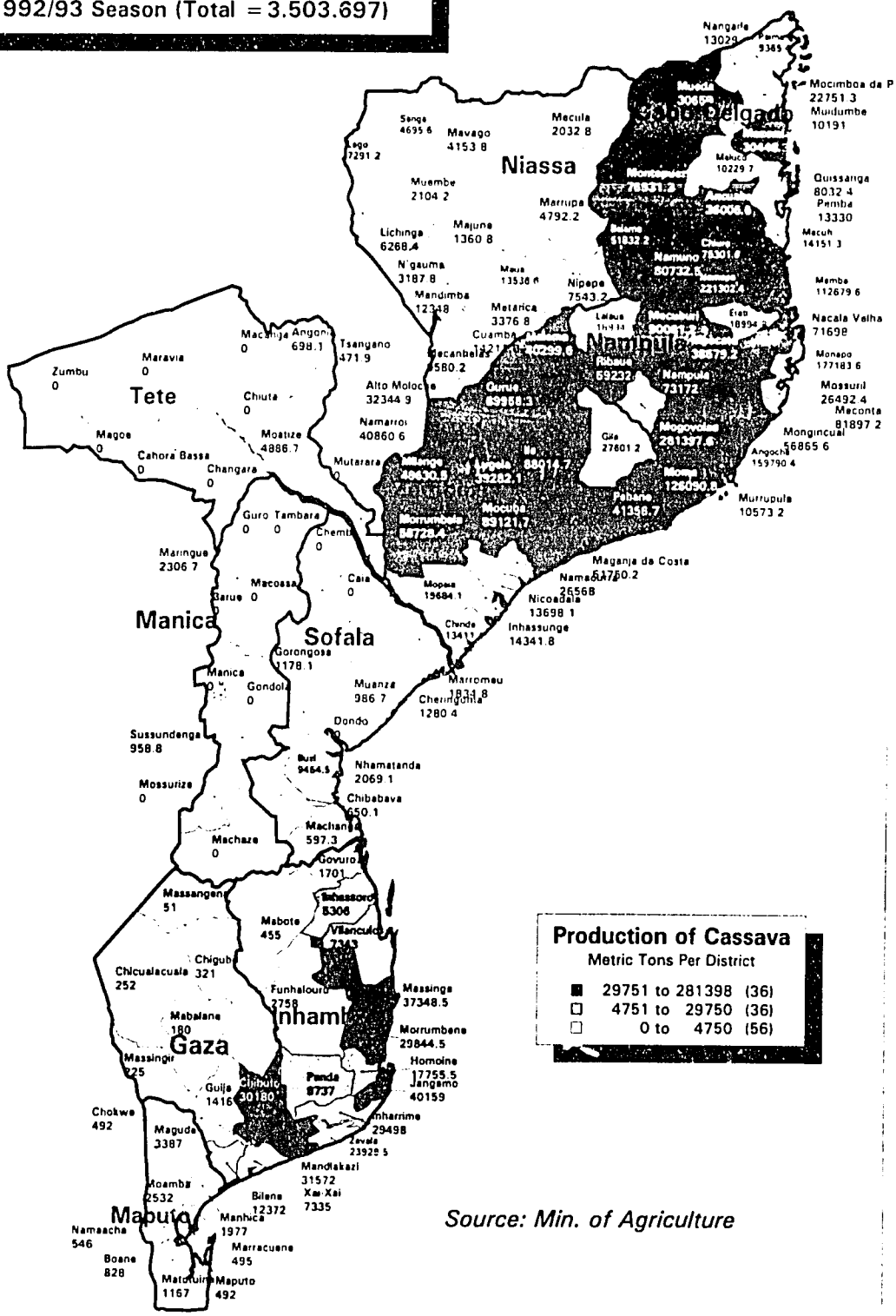


Source: U.S. State Department

303



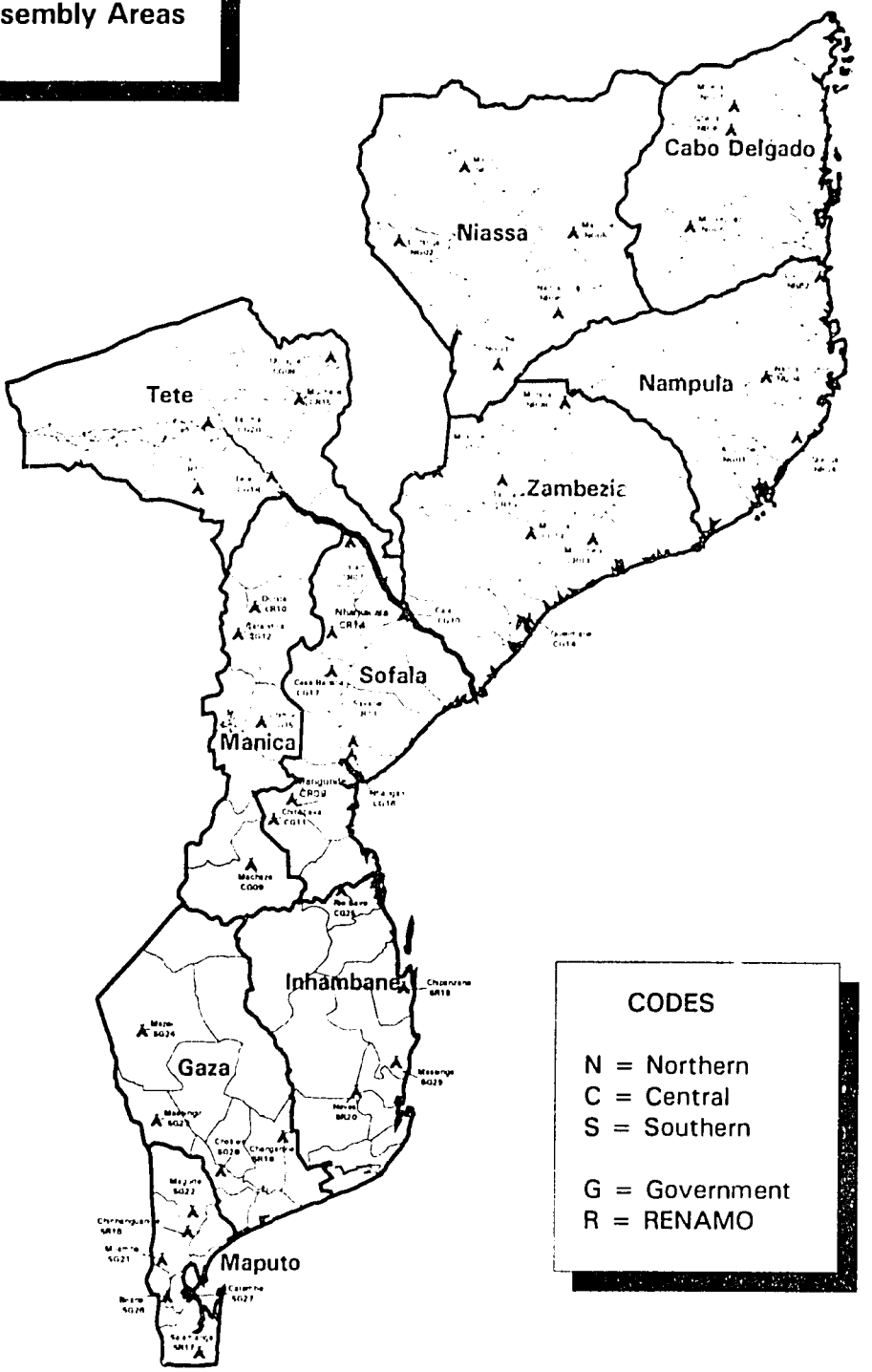
Estimated Family Production of Cassava
1992/93 Season (Total = 3.503.697)



United Nations for Humanitarian Assistance Coordination

December 1993

Assembly Areas



United Nations Office for Humanitarian Assistance Coordination November 1993

Appendix E

Sample Maps

Assembly Areas for Demobilization
Estimated Family Production of Cassava
Mine Incidents by District From 1975 to 1990
Mozambique Ethnic Groups circa 1960
Number of NGOs and Other Agencies Operating Programmes
Population Inside Mozambique: Estimates by District - Nov 1993
Population per Health Unit at Primary Level, by District
Projected Destination of 85,000 Demobilizing Soldiers
Republic of Mozambique