PN-ABG-754
69333

A COMPUTER SYSTEM FOR AID/REP PERFORMANCE INDICATORS

Purchase Order No. 306-0200-0-00-1000-CO

FINAL REPORT

Prepared for:

The Office of the AID Representative, Afghanistan
U.S. Agency for International Development
Peshawar, Pakistan

Prepared by:

Ron Barney
16 W. 72 St. # 15-D
New York, NY 10023
(212) 595-6235

November 8, 1990

Table of Contents

	Page
Contents	i
Overview	1
Introduction and Rationale	1
Purpose	1
Constraints	1
Scope of Work	2
Objectives and Indicators	2
Implications for Data Organization	4
Standardized Coding	4
Activity Code Development	4
Other Codes	5
Data Collection	6
Reporting Requirements	6
Data Requirements	7
Reports	7
Quarterly Indicators by Objective	7
Quarterly Indicators by Location	7
Organization of the Computer System	8
User Interface	8
Databases and Source Code	8
Enhancements	9
Future Reports	9
Utilities	9
Recommendations for Implementation	9
Scope of Work for Nathan-Berger	10

Appendixes

Data Requirements for Indicators by Activity Type	I-1
Input/Activity Codes	II-1
Report: Quarterly Indicators by Objective	III-1
Report: Quarterly Indicators by Location	IV-1
Data Dictionary	V-1
Source Code Documentation	VI-1

Overview

This report specifies details of a computer system for capturing and reporting quantified performance indicator data. Created under Purchase Order No. 306-0200-0-00-1000-00, the system is being delivered to AID/Rep's Data Collection and Analysis (DC&A) Unit.

A preliminary exposition of objectives, indicators and activity codes was presented by DC&A in October. Responses from Project Officers were addressed in a draft of the current report, issued in November. Concurrence on matters raised in the draft report has not yet been received from all Project Officers.

System implementation will be possible only after AID/Rep completes definition of its objectives and indicators, and only after the system has gone through a process of evaluation, modification and approval.

DC&A anticipates continuing development of the system under AID/Rep's Nathan-Berger studies contract. This report discusses the system and presents a draft scope of work for the Nathan-Berger study.

Introduction and Rationale

Purpose. Questions to be answered by D&A's consolidated data relate to where and how intensively Mission objectives are being supported as well as how well funds are reaching their intended beneficiaries, <u>not</u> how much is being spent on what by whom. Because the system is for management reporting, not accounting, it should allow activity measurements to be counted as supporting more than one objective. Also, pragmatic differences in reporting requirements can and should be made, so that large, long-term contractors submit data on an ongoing, quarterly basis, but smaller grantees with projects of less than a year report only at grant completion.

Constraints. There are several obstacles to setting up or maintaining even basic indicator reporting. One is the unique constraints of the Afghan situation, as set forth in AID/Rep's February 8, 1990 PPI submission. Standard measurements taken for granted in the United States--population figures and mortality rates, to name just two--are out-of-date or have never been collected; efforts to set up mechanisms to obtain them are frustrated by political chaos and cultural opposition.

Another is the difficulty of defining objectives and "doable" indicators given the situational constraints. As noted above, indicator definitions are is still not complete.

Yet another is the general lack of computer skills within the contractor community whose reporting is required. Most of AID/Rep's computerized contractors appear to work with systems that are imported, developed by now-absent temporary duty expatriate staff, or produced by local personnel who speak English as a second language and who therefore are at a disadvantage in learning or using English-based computer software.

Nevertheless, a rudimentary reporting system is in place, one set up to report O/AID/Rep activities by location. In order regularly to report indicator data in terms of current Mission objectives, a computerized data collection system must be imposed as a standard, across the board.

Scope of Work. Explicit requirements for the product being delivered are the following:

* Development of a computer system in compiled dBase III+ to collect baseline data required for quantitative monitoring of O/AID/Rep program progress indicators. The system will allow collection of baseline data and the processing of specific beneficiary monitoring reports. The system will be compatible with other data systems for Afghanistan developed by the Mission's DC&A unit and systems developed by other international donor/relief organizations.

* In cooperation with Project Officers, determination of those indicators which are appropriate for quantitative monitoring of progress achieved toward program objectives, and obtainable from the field.

* In cooperation with O/AID/Rep's contractors and grantees in Peshawar, creation of a standard coding system for computerized data analysis.

* Development of a set of recommendations for data collection procedures to follow up initial organization of the system.

* Development of a scope of work for follow-on activities to be undertaken by Nathan/Berger under the Afghanistan Studies component of the CEP project.

Objectives and Indicators

Since the October presentation of draft objectives and indicators, some changes to objectives and indicators have been submitted. The following revision takes these submissions into consideration.

1. Address food security through programs aimed at restoring agricultural production and markets and the short-term provision of food to deficit areas and vulnerable groups.

Indicators:

A. Prices of basic foodstuffs and cash crops produced in liberated areas, as proxy indicators of relative scarcity and availability.

B. Types, amounts and values of donated emergency commodities delivered inside Afghanistan.

C. Types, amounts and values of traded commodities delivered inside Afghanistan.

D. Types, amounts and values of physical infrastructure created or rehabilitated inside Afghanistan.

2. Assist in providing ultimately sustainable health and education services to Afghans remaining in or returning to their country.

Indicators:

E. For each health and education activity, the total of operating costs (defined either as salaries, pharmaceuticals and supplies or as salaries, books and supplies) funded by USAID.

F. For each health and education service, the number of Afghans providing professional services, the number of non-Afghans providing these services, and the Afghan number expressed as a percent of the non-Afghan total.

3. Reduce the production of opium in Afghanistan, and increase awareness of the negative effects of narcotics production among target groups.

Indicators:

G. Levels of poppy production in monitored areas of Afghanistan, as a proxy indicator for poppy production in the country as a whole.

H. Number of persons trained in narcotics awareness among target groups.

4. Assist with strengthening human capital and institutional capabilities by increasing equitable access to AID-supported training.

Indicators:

I. Number of Afghans (broken out by gender) trained in AID-financed administration, financial management and vocational skills programs.

J. For each AID-supported Afghan organization, the share of administrative and technical services provided by Afghan personnel, the share provided by non-Afghans, and the Afghan amount expressed as a percent of the total, as an indicator of improved administrative and technical capacity.

5. Strengthen effective representative local and national Afghan organizations and groupings in support of a more open and less insular society.

Indicators:

K. Level of financial assistance provided through Afghan NGOs.

L. Number of Afghan NGOs assisted.

M. Number of workshops, seminars or conferences with democratic pluralism themes sponsored.
N. Number of Afghans participating in international forums.

Implications for Data Organization. A simple way of organizing indicator data is to set up a single file structure for all the kinds of information to be collected, filling in those parts that are pertinent and leaving others blank. Such a file structure is shown in Appendix I. The left-hand column shows categories of information to which all of the items in the same row apply. Subsequent columns group AID/Rep activities according to the indicator(s) to which they pertain, and specify more exactly the information required in each horizontal category.

No provision has been made in the column setup for indicators K and L. Because these can be easily derived separately, data need not be gathered for them as part of the system.

One of the horizontal categories, labelled as "Supports DPI?" refers to democratic pluralism initiatives, which must be handled as a special case. Assignment of items to DPI must be explicit; thus it requires a separate field.

Further discussion of the horizontal categories is presented later in this paper, in the section on organization of the computer system.

Standardized Coding

There are two main reasons for a code-intensive approach to computer system design. One is that it makes the storage requirements of the system smaller. The other is exactness, a requirement for machine processing.

Five things need to be coded in order to identify any indicator measurement for AID/Rep:

* type of input
* type of indicator toward which this input contributes
* location of the input
* contractor performing the activity/input
* if this is not the only input of this kind undertaken by this contractor in this place, serial identification of which one it is.

All coded references to indicator measurements must be standardized, and as subsidiary information, AID/Rep also needs standard codes identifying associated Afghan political parties.

Input/Activity Code Development. The most difficult problem involved in developing the system is that of creating a classification of AID/Rep inputs/activities which, through a corresponding set of codes, will allow unique identification of every input/activity.

In studying this problem, USAID activity codes used in accounting, sector codes used by ACBAR, UNDP and UNIDATA, and sector/accounting codes used by UNHCR were all examined. Analysis of AID/Rep activities showed that none of these listings was comprehensive enough to cover current AID/Rep inputs. Moreover, all of the classification systems seemed to have been put together piecemeal.

The solution chosen was to create a new coding scheme. This is presented in Appendix II with revisions based on comments received after presentation of the first draft. The revised coding scheme has several important features:

* Category classifications are limited to four levels. More levels would allow finer distinctions, but four seem to be adequate.
* Information at the fourth level may be aggregated at the third level for reporting purposes.
* Code construction--in numbered levels--is parallel to that of the 1975 Provisional Gazetteer of Afghanistan, whose location codes are standardized for AID/Rep use. This allows the user interface to be simplified by programming a single approach to code selection.
* The first three levels are fixed, but the fourth level is intended to be expandable: it is displayed alphabetically, not hierarchically or conceptually, so that new items can be easily inserted. New items at the fourth level can be keyed with one code number but several names (e.g. dal chana, chick peas, garbanzos) where this is appropriate.

Other Codes. Other coding to be standardized within the system is far less problematic. Only one element, location coding, needs significant future enhancement.

* Indicator Codes. As was noted above, a single measurement may be counted as an indicator of more than one objective. Codes attached to measurement records identify the objectives each should be reported as supporting. The codes can change as objectives change.

  Indicators can be assigned by DC&A without user involvement, eliminating the possibility of user error.

  In the special case of activities/inputs supporting democratic pluralism, the breadth of potential activity requires explicit assignment. Not every conference has a democratic pluralism theme. If one does, it must be given a democratic pluralism indicator code.
* Location Codes. The Gazetteer codes developed by Tom Eighmy are already required by AID/Rep for province and district identification within Afghanistan. A digitized version of province, district and roughly 22,000 village codes from the Gazetteer have been produced at AVICEN and UNDP under Roger Helms. About 8,000 subvillages have yet to be added to the database.

No attempt is being made at this time to add detailed location codes for AID/Rep activities taking place in Pakistan, though the level of these activities may be significant.

* Contractor Codes. There appears to be no simple, fixed, standard accounting code for AID/Rep contractors. A two-digit number has been arbitrarily assigned to each contractor for this purpose.
* Sequence Codes. These numbers distinguish similar activities undertaken by a contractor in a given place. They are necessary because, for instance, three karezes might be cleaned in a specific village, and two BHWs might work there independently of each other. The numbers are automatically assigned at record creation.
* Party Codes. For monitoring purposes DC&A needs to know the political aegis under which activities take place--a party-affiliated commander, a party-dominated shura or a shura without known party lines. A simple numbering system has been imposed to standardize coding of this information.

Data Collection

Reporting Requirements. At least four aspects of the system's reporting requirements will be at variance with current practice or with recent discussion.

* Normal Reporting Period. Data must continue to be reported by contractors on a quarterly basis, both so that it can be checked on a continuing basis and so that it can be aggregated into years either on a fiscal or calendar basis.
* Special Consideration for Small Grants. As a help to small contractor/grantees, DC&A will require only a single set of data for grants which are valued at less than $500,000 and last less than one year. All indicators for such grants will be recognized in the quarter of grant completion.
* Non-Cumulative Facility Costs. Departing from present DC&A practice, costs of supporting educational and health facilities will be reported on a current-period basis and not as a cumulative total beginning with facility establishment. This change is useful in setting up multiquarter reports; and it is necessary to make reporting of facility support consistent with other data.
* Written Definitions. Contractors will provide written definitions of dollar amounts (except indicator amounts for market prices) included in each set of data. In order to facilitate reporting, exchange rates to be used will be those prevailing on the last day of the quarter being reported. The intention of this requirement is to capture the value reaching the field not counting transportation costs or the costs of administrative

overhead incurred outside Afghanistan. In most cases the cost of inputs will simply be the contractor's purchase price.

Contractors will also provide written definitions of their methods of estimating indirect beneficiaries, for activity indicators requiring this number.

Data Requirements. Software for the AID/Rep Performance Indicators system--including an automatic installation program--is contained on three, 720-K, 3-1/4" diskettes and is ready to be distributed to contractors/grantees for testing. The system requires an IBM-compatible microcomputer with approximately 5MB of available hard disk storage, DOS operating system version 3.3 or higher. Minor expansion of the normal CONFIG.SYS and AUTOEXEC.BAT files is explained in a READ.ME file on the first installation diskette. In its initial version the software supports only the Epson LQ-1050 printer, but this printer is not essential for contractor input.

Source code dBase III+ programs with Clipper modifications is printed in Appendix VI and is being delivered to DC&A on diskette. It can be made available to contractors who wish to tailor the system to other uses.

The software allows contractors to produce database records conforming to AID/Rep specifications, and download the database to diskette for submission to DC&A. It allows DC&A to upload files contractor data from diskette to an accumulative main database, and print reports.

Reports

The two reports attached as Appendixes III and IV contain fictitious data, and are shown solely to demonstrate the format of reports available from the system. Both reports are printable at will.

Quarterly Indicators by Objective takes each objective and indicator in turn (except where two indicators have similar reporting for the same objective), and summarizes by location all indicator measurements that have been flagged with the current indicator code.

Note that the order of printing follows that of the indicators and activity codes: the list of indicators and the structure of the activity codes determines structure of the reports.

Data is grouped at the third activity code level, though specific measurements at the fourth level are shown. All of the coding used in keying in the indicators is dropped from the report as unnecessary detail. One indicator, the number of Afghan NGOs assisted, is not included in this report because it can be easily counted by hand.

Quarterly Indicators by Location takes in turn each district for which there are current indicators, and summarizes indicators for the district by objective. Health and education facilities for each district are totalled and printed on the line

identifying the type of facility. In addition, monitoring status of activities is displayed in columnar groups at the right edge of the page, and activities of each monitoring status are summed for display with indicator, district and province totals.

Organization of the Computer System

User Interface. As has been stated above, general lack of computer skills in the contractor community demands a system that is easy-to-use. Several requirements of the system have been inferred from this demand.

* Menu-Driven Operation. The system is self-contained, so that all operations take place through commands executed from function menus displayed on screen. Contractors are able not only to key in data, but also to download that data to diskette without leaving the software. DC&A, likewise, should be able to upload contractor data to the main database as part of the software before indexing the information and running reports.
* On-Screen Code Lookup. Complexity of the coding schemes makes it essential that coding choices be made through reference lists shown on screen. The alternative, looking up coding choices from paper lists, is extraordinarily time-consuming and error-prone. The system contains all valid coding possibilities.
* Built-in Code Validation. Any code entered from an on-screen listing is checked against system records and the meaning of that code shown on screen for confirmation.
* Optional Coding Defaults. The user is able to limit the system so that factors that remain constant in use may be set as constants in the system. Constants which may be determined through a menu selection include the current quarter, current year, contractor identification, and whether the system is to display prices or units (for instance, afghanis per seer or metric tons).
* Optional Field Suppression. The part of each record showing allowing entry of a subgrant number will not be used by all contractors. The user can suppress display of this field as a coding default.
* Indicator-Driven Field Masking. As is shown in Appendix I, most activities fall into groups that may be implicitly associated with specific indicators; and those indicators have patterns of information use. In order to simplify input, the implied indicator of any given code determines indicator data fields will be seen on screen. Unused fields are masked from view.

Databases and Source Code. A simple data dictionary showing database field names and their meanings is given in Appendix V. More technical data definitions are given in the source code documentation of Appendix VI.

Enhancements

The system described above should meet AID/Rep's immediate needs. Other system features have been discussed during the design process which might be added in the future.

Future Reports

* Indicators by Contractor/Grantee. Within the listing of contractor/grantees, data would probably be listed by location.
* Multiperiod Summary of Indicators by Objective. The most likely summaries would be by fiscal or calendar year. Totals of indicators by type should be printable for any set of beginning and ending periods.
* Multiperiod Summaries of Indicators by Location. As with the multiperiod summaries of indicators by objective, the reports most likely to be needed would be those by fiscal or calendar year.

Utilities

* Activity Code Editing. However carefully the activity code listing is prepared, needs for modification will arise. A utility needs to be written to allow changes to the code databases. DC&A can then make the changes and distribute revised files to contractors, or authorize contractors to make specific changes.
* System Backup. Particularly because power outages can damage database integrity, a utility needs to be added which will allow DC&A to back up the main database and, if necessary restore it from diskette.
* Partial Reindexing. Depending on the machine being used, indexing of the system even at its present, dataless stage can take fifteen to twenty minutes. The reindexing utility needs to be rewritten to allow reindexing of only those parts of the system that have been changed--for instance, activity codes. A call for large-scale reindexing should display a warning and require confirmation from the user before proceeding.

Recommendations for Implementation

The system as delivered is preliminary, needing further review and development before it can be implemented. Because objectives and indicators have not yet been fully defined, AID/Rep's commitment to 1990 as baseline year for data should be reevaluated: 1991 might be more appropriate, with the first data produced on the system being that of the April-June quarter (deliverable to DC&A at the end of July). Follow-on activities through Nathan-Berger, even if greatly expedited, might not begin before March 1991 because they cannot be budgeted before January.

Scope of Work for Nathan-Berger. Some of the work of system refinement may be accomplished within the United States. In order to make possible implementation in the April-June quarter, Nathan-Berger will need to begin work for DC&A as soon as possible, sending a team to Peshawar no later than mid-March. The following list of tasks need to be performed as part of the firm's scope of work:

* Obtain concurrence on Mission objectives and indicators or, if concurrence is delayed past, say, January 31, obtain a ruling on these items from the Mission head.
* Provide review of activity codes by a professional library scientist familiar with the conceptual problems of classification scheme development. Since the activity codes were developed almost entirely in terms of current program, the review should also encompass general USAID mandates as a way of broadening the classifications for future program. "Choice" and "governance" activities under DPI should be addressed, as should "open markets" and the environmental initiative announced earlier this year.
* Invite a limited-time-only review of activity codes through the Agency Coordinating Body for Afghan Relief (ACBAR), Coordination of Medical Committees (CMC) and World Health Organization (WHO). ACBAR has a broad-ranging coding scheme of its own; the three organizations working together have developed a classification of health facilities that should not be ignored. Such a review may uncover flaws in the coding that would not otherwise be revealed before implementation.
* Recommend a formal procedure for proposing, reviewing and implementing changes to activity codes once the system is in place. Such a procedure is necessary to maintain usefulness and uniformity within the user community.
* Finalize with contractors and Project Officers the ways activities will be measured. The product as delivered has draft measures in place (<u>metric tons</u> of wheat delivered; <u>afghanis/seer</u> of wheat priced), but these have been assigned on a superficial basis and may not meet reporting needs.
* Expand the <u>Gazetteer</u> village database to include approximately 8,000 <u>sub</u>villages that have not yet been input.
* Determine the level to which <u>Gazetteer</u> listings should be expanded to include places in Pakistan, research whether these listings are already available from government sources in electronic form, and expand the location databases.
* Modify the user interface of the system on an as-needed basis to make it easier to use.
* Enhance system capabilities by creating the reports and utilities described above, and by creating other reports that may be required.
* Write and duplicate a user manual for the final system.

* Distribute copies of the software and user manual to contractor/grantees.
* Offer training and limited programming assistance to contractor/grantees wishing to integrate the system with their existing software or wishing to write programs to convert their existing data to that of the system.
* Working with the contractor/grantees, help translate data from the January-March quarter to the format of the new system.

Contractor diskettes for April-June would be due at DC&A on July 31.

# # #

DATA REQUIREMENTS FOR INDICATORS BY ACTIVITY TYPE

DATA FIELD NAME	Objective 1 Indicator A TYPE 1	Objective 1 Indicator B TYPE 2	Objective 1 Indicator C TYPE 3	Objective 1 Indicator D TYPE 4 ***	Objectives 2, 4 Indicators E, F, J TYPE 5 ****	Objective 3 Indicator G TYPE 6	Objective 3 Indicator H TYPE 7	Objective 4 Indicator I TYPE 8 ***	Objective 4 Indicator J TYPE 9 ***	Objective 5 Indicator M TYPE 10 ***	Objective 5 Indicator N TYPE 11 ***
ACTIVITY TYPE (to be specified by code)	Basic foods and cash crops priced	Donated emergency commodities	Traded emergency commodities	Infrastructure created/ rehabilitated	Health facility/ school support	Opium production	Narcotics Awareness training	Non-school skills training	AID-supported Afghan Institutions *****	DPI meetings	Afghans in/ at Intl. Forums
QUANTITY OR PRICE	Market price	Quantity	Quantity	Quantity	Quantity	Quantity	Quantity	Quantity	NA	Quantity	Quantity
UNITS OF MEASUREMENT	*	*	*	*	Health Fac./School	*	*	*	NA	*	Persons
$ FROM USAID	NA	Value delivered	Value, items traded **	Value, items completed **	Operating costs	NA	Cost of training	NA	$ support	Cost	Cost
TRANSPORTATION COST	NA	*	*	*	*	NA	NA	NA	NA	NA	NA
BENEFICIARIES:											
SALARIED MALE AFGHANS	NA	S	S	S	Health workers/ teachers	NA	NA	Trainees	S	NA	NA
SALARIED FEMALE AFGHANS	NA	S	S	S	Health workers/ teachers	NA	NA	Trainees	S	NA	NA
SALARIED NON-AFGHANS	NA	S	S	S	Health workers/ teachers	NA	NA	NA	S	NA	NA
UNSALARIED DIRECT	NA	*	*	*	Students/patients *	NA	[illegible]	NA	NA	Participants	Participants
INDIRECT	NA	*	[illegible]	*	NA	NA		NA	NA	*	*
SUPPORTS DPI?	NA	NA	NA	NA	NA	NA	NA	NA	NA	*	*
IDENTIFICATION AND MONITORING INFORMATION:											
PROVINCE CODE	S	S	S	S	S	S	S	S	S	S	S
DISTRICT CODE	S	S	S	S	S	S	S	S	S	S	S
VILLAGE CODE	S	S	S	S	S	S	S	S	S	S	S
CONTRACTOR CODE	S	S	S	S	S	NA	S	S	S	S	S
CONTRACTOR SEQUENCE CODE	A	S	S	S	S	NA	S	S	S	S	S
SUBCONTRACT	NA	S	S	S	S	NA	S	S	S	S	S
PERIOD REPORTED	S	S	S	S	S	S	S	S	S	S	S
DATA ENTRY DATE	S	S	S	S	S	S	S	S	S	S	S
PROJECT BEGIN DATE	S	S	S	S	S	NA	NA	S	S	S	S
PROJECT END DATE	S	S	S	S	S	NA	NA	S	S	S	S
VERIFICATION STATUS	NA	S	S	S	S	NA	NA	NA	NA	NA	NA
VERIFICATION DATE	NA	S	S	S	S	NA	NA	NA	NA	NA	NA
PARTY AFFILIATION	NA	S	S	S	S	NA	NA	NA	NA	NA	NA
COMMANDER	NA	S	S	S	S	NA	NA	NA	NA	NA	NA

* - to be defined by the appropriate Project Officer
** - expected to include food portion of Food for Work projects. To be defined by Project Officer
*** - contracts of less than one year's duration to be reported only at contract completion
**** - beneficiary information to be associated with salary data only
***** - other than schools and health facilities
NA - not applicable
S - standard information required where applicable

1 **Relief**
- 1.1 Commodities
  - 1.1.1 Food
    - 1.1.1.1 Chickpea (dal chana, garbanzo)
    - 1.1.1.1 Garbanzo (chickpea, dal chana)
    - 1.1.1.1 Dal chana (chickpea, garbanzo)
    - 1.1.1.2 Ghee, vegetable
    - 1.1.1.3 Kidney bean (lubia)
    - 1.1.1.3 Lubia (kidney bean)
    - 1.1.1.4 Salt
    - 1.1.1.5 Sugar
    - 1.1.1.6 Tea, black
    - 1.1.1.7 Wheat
  - 1.1.2 Commodity substitutes
    - 1.1.2.1 Cash for food
  - 1.1.3 Operating costs
    - 1.1.3.1 Field salaries
- 1.2 Shelter
- 1.3 Medical care/sanitation
  - 1.3.1 Program
    - 1.3.1.1 McCollum flights
- 1.4 Relocation

2 **Safety**
- 2.1 Mine clearance
  - 2.1.1 Dog deployment
    - 2.1.1.1 Roads cleared
  - 2.1.2 Operating costs
    - 2.1.2.1 Field salaries
- 2.2 Human resource development
  - 2.2.1 Training
    - 2.2.1.1 Mine detection dog handling
  - 2.2.2 Operating costs
    - 2.2.2.1 Field salaries

3 **Infrastructure**
- 3.1 Structures
  - 3.1.1 Environmental modeling structures
    - 3.1.1.1 Gabion
    - 3.1.1.2 Levee
    - 3.1.1.3 Terrace
    - 3.1.1.4 Wall, retaining
  - 3.1.2 Water resource, delivery and removal structures
    - 3.1.2.1 Aqueduct
    - 3.1.2.2 Canal
    - 3.1.2.3 Culvert
    - 3.1.2.4 Dam
    - 3.1.2.5 Ditch (jui)
    - 3.1.2.6 Flume
    - 3.1.2.7 Intake
    - 3.1.2.8 Jui (ditch)
    - 3.1.2.9 Kariz
    - 3.1.2.10 Reservoir
    - 3.1.2.11 Sluice gate
    - 3.1.2.12 Spring

3.1.2.13 Syphon
3.1.2.14 Tunnel
3.1.2.4 Weir
3.1.2.15 Well
3.1.3 Buildings
3.1.3.1 Government building
3.1.3.2 House
3.1.3.3 Medical building
3.1.3.4 Mosque
3.1.3.5 School
3.1.3.6 Store house
3.1.4 Transporation infrastructure
3.1.4.1 Airport
3.1.4.2 Bridge, Bailey
3.1.4.3 Causeway
3.1.4.4 Crossing, Irish (wash lining)
3.1.4.5 Railway
3.1.4.6 Road, main
3.1.4.7 Road, national highway
3.1.4.8 Road, rural (farm to market)
3.1.5 Power/electrification
3.1.5.1 Hydroelectric generator
3.1.5.2 Wind-driven generator
3.1.6 Raw materials/parts
3.1.6.1 Cement
3.1.6.2 Chicken wire
3.1.6.3 Roof beams, concrete
3.1.6.4 Roof sheeting, plastic
3.1.6.5 Steel
3.1.7 Production tools/fuel
3.1.7.1 Earth moving equipment (list...)
3.1.7.2 Trade tools, auto mechanic (list...)
3.1.7.3 Trade tools, carpenter (list...)
3.1.8 Maintenance
3.1.9 Operating costs
3.1.9.1 Field salaries
3.2 Human resource develo nent
3.2.1 Training
3.2.1.1 Carpentry
3.2.1.2 Construction supervision
3.2.1.3 Electricity
3.2.1.4 Engineering, Construction
3.2.1.5 Engineering professional development course for degree holders
3.2.1.6 Engineering refresher course for degree holders
3.2.1.7 Masonry
3.2.1.8 Mathematics
3.2.1.9 Plumbing
3.2.1.10 Steel work
3.2.1.11 Welding
3.2.2 Conferences/workshops
3.2.2.1 Support for events
3.2.3 Participation in international forums
3.2.3.1 Support for individuals

3.2.4 Operating costs
3.2.4.1 Field salaries

4 **Production**
4.1 Mineral Production
4.2 Biological Production
4.2.1 Plants and plant products
4.1.1.1 Apple
4.1.1.2 Apricot
4.1.1.3 Chickpea (dal chana/garbanzo)
4.1.1.3 Dal chana (chickpea/garbanzo)
4.1.1.4 Cotton
4.1.1.5 Cucurbit
4.1.1.3 Garbanzo (chickpea/dal chana)
4.1.1.6 Gourd
4.1.1.7 Grape
4.1.1.8 Green pea (mutter)
4.1.1.9 Kidney bean (lubia)
4.1.1.10 Lentil
4.1.1.11 Licorice
4.1.1.9 Lubia (kidney bean)
4.1.1.12 Maize
4.1.1.13 Melon
4.1.1.14 Mulberry
4.1.1.8 Mutter (green pea)
4.1.1.15 Olive
4.1.1.16 Peach
4.1.1.17 Plum
4.1.1.18 Poplar
4.1.1.19 Poppy, opium
4.1.1.20 Raisin
4.1.1.21 Rice
4.1.1.22 Wheat
4.2.2 Animals and animal products
4.2.2.1 Carp
4.2.2.2 Chicken
4.2.2.3 Chicken eggs, fresh
4.2.2.4 Cow, dairy
4.2.2.5 Duck
4.2.2.6 Horse
4.2.2.7 Mule
4.2.2.8 Mutton, fresh
4.2.2.9 Oxen
4.2.2.10 Sheep
4.2.3 Chemical Inputs/Feed
4.2.3.1 DAP
4.2.3.2 NPK
4.2.3.3 Urea
4.2.4 Production tools/fuel
4.2.4.1 Fish net
4.2.4.2 Fuel, diesel
4.2.4.3 Fuel, kerosene
4.2.4.4 Pick axe
4.2.4.5 Plow
4.2.4.6 Saw, hand

4.2.4.7 Shears, wool
4.2.4.8 Shovel
4.2.4.9 Sickle
4.2.4.10 Thresher, animal powered
4.2.4.11 Thresher, machine powered
4.2.4.12 Tractor, 45 horse power
4.2.4.13 Tractor, 75 horse power
4.2.4.14 Tractor harrow
4.2.4.15 Tractor plow
4.2.4.16 Tractor soil leveling blade
4.2.4.17 Tractor trailer
4.2.4.18 Tractor tine tiller
4.2 4.19 Tractor spare part
4.2.4.20 Trade tools, flour miller
4.2.4.21 Trade tools, blacksmith
4.2.4.22 Water pump, hand
4.2.4.23 Water mill
4.2.4.24 Wheelbarrow
4.2.4.25 Wind mill

4.2.5 Health care
4.2.5.1 Antifungal chemicals (list)
4.2.5.2 Locust control chemicals (list)
4.2.5.3 Para-Veterinary medicine kit
4.2.5.4 Pesticides, plant (list)
4.2.5.5 Vaccinations (list)

4.2.6 Animal training
4.2.6.1 Traction training for mules

4.2.7 Operating costs
4.2.7.1 Field salaries

4.3 Manufacturing Industries

4.3.1 Textiles
4.3.1.1 Cottage industry

4.3.2 Operating costs
4.3.2.1 Field salaries

4.4 Service Industries

4.4.1 Employment
4.4.1.1 Employment exchange

4.4.2 Operating costs
4.4.2.1 Field salaries

4.5 Trade development.

4.5.1 Credit assistance
4.5.1.1 Letter of credit

4.5.2 Bazaar development
4.5.2.1 Export marketing
4.5.2.2 Product specification assistance
4.5.2.3 Negotiation assistance
4.5.2.4 Vendor identification

4.5.3 Transportation assistance
4.5.3.1 Transportation clearances
4.5.3.2 Transportation rebates

4.5.4 Operating costs
4.5.4.1 Field salaries

4.6 Human resource development

4.6.1 Training/Extension
4.6.1.1 Agricultural marketing agent training

4.6.1.2 Demonstration farm
4.6.1.3 Blacksmith training
4.6.1.4 Knitting machine training
4.6.1.5 Para-veterinary medicine
4.6.1.6 Tractor repair
4.6.1.7 Office skills training
4.6.1.8 Spare parts supply and storage training
4.6.2 Conferences/workshops
4.6.2.1 Support for events
4.6.3 Participation in international forums
4.6.3.1 Support for individuals
4.6.4 Operating costs
4.6.4.1 Field salaries

5 **Health** (other than relief health activities)
5.1 Preventive medicine facility support
5.1.1 Immunization cold chains
5.1.1.1 Field salaries
5.1.1.2 Pharmaceuticals
5.1.1.3 Supplies and medical equipment
5.1.2 MCH (Maternal and Child Care) posts
5.1.2.1 Field salaries
5.1.2.2 Pharmaceuticals
5.1.2.3 Supplies and medical equipment
5.1.3 MCH clinics
5.1.3.1 Field salaries
5.1.3.2 Pharmaceuticals
5.1.3.3 Supplies and medical equipment
5.1.4 Malaria programs
5.1.5 Tuberculosis prevention programs
5.2 Clinical care facility support
5.2.1 Basic health post
5.2.1.1 Field salaries
5.2.1.2 Pharmaceuticals
5.2.1.3 Supplies and medical equipment
5.2.2 Basic health center (clinic)
5.2.2.1 Field salaries
5.2.2.2 Pharmaceuticals
5.2.2.3 Supplies and medical equipment
5.2.3 Comprehensive health center
5.2.3.1 Field salaries
5.2.3.2 Pharmaceuticals
5.2.3.3 Supplies and medical equipment
5.2.4 Primary care hospital
5.2.4.1 Field salaries
5.2.4.2 Pharmaceuticals
5.2.4.3 Supplies and medical equipment
5.2.5 Provincial hospital
5.2.5.1 Field salaries
5.2.5.2 Pharmaceuticals
5.2.5.3 Supplies and medical equipment
5.2.6 Regional hospital
5.2.6.1 Field salaries
5.2.6.2 Pharmaceuticals
5.2.6.3 Supplies and medical equipment

5.2.7 Training facilities in Pakistan
5.2.7.1 Field salaries
5.2.7.2 Pharmaceuticals
5.2.7.3 Supplies and medical equipment
5.3 Human resource development
5.3.1 Training
5.3.1.1 Advanced Medical Assistant Training
5.3.1.2 BHW Refresher
5.3.1.3 Cold chain managers
5.3.1.4 Cold chain technician (CCT)
5.3.1.5 Community Developer/Health Educator
5.3.1.6 Female Health Workers (FHW, or dai)
5.3.1.7 Immunization technician (IT)
5.3.1.8 Laboratory Technician training
5.3.1.9 Male Basic Health Workers (BHWs)
5.3.1.10 MCH broadcasts
5.3.1.11 MCH clinic planning and training workshop
5.3.1.12 MCH courses
5.3.1.13 Nurse Inservice Training
5.3.1.14 Nurse-Midwife training
5.3.1.15 Nurse Refresher
5.3.1.16 Outreach Worker (ORW) Course
5.3.1.17 Paramedic
5.3.1.18 Paramedic, Advanced
5.3.1.19 Paramedic, Expanded
5.3.1.20 Paramedic, Refresher
5.3.1.21 Physician Refresher Course
5.3.1.22 Physician Retraining
5.3.1.23 Public Health Worker (PHW)
5.3.1.24 Rural Health Officers (RHO)
5.3.1.25 Teaching Clinic sessions
5.3.1.26 Training Evaluator
5.3.1.27 Training of Trainers, for physicians
5.3.1.28 Vaccinator
5.3.1.29 X-ray Technician
5.3.2 Conferences/workshops
5.3.2.1 Support for events
5.3.3 Participation in international forums
5.3.3.1 Support for individuals
5.3.4 Operating costs
5.3.4.1 Field salaries

6 **Education** (other than work-related skills training)
6.1 Support for individual higher education
6.1.1 Completion of post-secondary education (Weber Scholarship Program)
6.1.1.1 Support for individuals
6.1.2 Skills upgrading (Afghan Scholarship Program)
6.1.2.1 Support for individuals
6.2 Short-term educational programs (other than work-oriented skills training and public information campaigns)
6.2.1 Adult Literacy
6.2.1.1 Field salaries
6.2.1.2 Supplies
6.2.1.3 Texbooks

7.3.4 Operating expense
7.3.4.1 Field salaries

8 Expression of ideas/Culture/Pure sciences
8.1 Media support
8.1.1 Publications
8.1.1.1 Magazines
8.2 Human resource development
8.2.1 Conferences/workshops
8.2.1.1 Support for events
8.2.2 Participation in international forums
8.2.2.1 Support for individuals
8.2.3 Operating costs
8.2.3.1 Field salaries

6.3 Educational facility support
- 6.3.1 Primary schools (lower grades, usually beginning with grade 1)
  - 6.3.1.1 Field salaries
  - 6.3.1.2 Supplies
  - 6.3.1.3 Textbooks
- 6.3.2 Secondary schools (upper grades, usually beginning with grade 7 and ending with grade 12)
  - 6.3.2.1 Field salaries
  - 6.3.2.2 Supplies
  - 6.3.2.3 Textbooks
- 6.3.3 Post-secondary facilities (beyond grade 12 or its equivalent)
  - 6.3.3.1 Field salaries
  - 6.3.3.2 Supplies
  - 6.3.3.3 Textbooks

6.4 Human resource development
- 6.4.1 Training
  - 6.4.1.1 General teacher training
  - 6.4.1.2 Supervisory teacher training
  - 6.4.1.3 Teacher training for adult literacy
- 6.4.2 Conferences/workshops
  - 6.4.2.1 Support for events
- 6.4.3 Participation in international forums
  - 6.4.3.1 Support for individuals
- 6.4.4 Operating costs
  - 6.4.4.1 Field salaries

7 Public Policy and Administration

7.1 Public information
- 7.1.1 Information gathering
  - 7.1.1.1 Field survey
  - 7.1.1.2 Satellite information
- 7.1.2 Computerized information processing
- 7.1.3 Public information campaigns
  - 7.1.3.1 Narcotics Awareness Training
- 7.1.4 Operating costs
  - 7.1.4.1 Field salaries

7.2 Government institution support
- 7.2.1 ECA
  - 7.2.1.1 Office expense
  - 7.2.1.2 Salaries
- 7.2.2 MOPH
  - 7.2.2.1 Office expense
  - 7.2.2.2 Salaries
- 7.2.3 RAFA
  - 7.2.3.1 Office expense
  - 7.2.3.2 Salaries

7.3 Human resource development
- 7.3.1 Training
  - 7.3.1.1 Public administration training
- 7.3.2 Conferences/workshops
  - 7.3.2.1 Support for events
- 7.3.3 Participation in international forums
  - 7.3.3.1 Support for individuals

OBJECTIVE: Food Security
INDICATOR: Market Prices for Basic Foods

Item	Origin	Amount	Units of Measurement	Dollars USAID Contribution	Dollar Transportation Cost	Beneficiaries: Salaried: Afghan Males	Afghan Fem.	Non-Afgh.	Total Salaried	Afgn. % of Total	Unsalaried	Indirect	Project Dates: Start	End	Monitoring Status/Date
BADAKSHAN PROVINCE															
Baharak District															
Production/Biological production/Plants and plant products															
Chickpea (dal chana/garbanzo)		20	afghanis/seer												
Chickpea (dal chana/garbanzo)		22	afghanis/seer												
Kidney bean (lobia)		15	afghanis/seer												
Rice		9	afghanis/seer												
Wheat		10	afghanis/seer												
Production/Biological production/Animals and animal products															
Chicken		25	afghanis/seer												
Chicken eggs, fresh		17	afghanis/seer												
Mutton, fresh		20	afghanis/seer												
District Totals				0	0	0	0	0	0		0	0			
Zebak District															
Production/Biological production/Animals and animal products															
Chicken		[illegible]	afghanis/seer												
Chicken eggs, fresh		20	afghanis/seer												
Mutton, fresh		30	afghanis/seer												
District Totals				0	0	0	0	0	0		0	0			
PROVINCE TOTALS				0	0	0	0	0	0		0	0			
ZABUL PROVINCE															
Arghandab District															
Production/Biological production/Plants and plant products															
Chickpea (dal chana/garbanzo)		[illegible]	afghanis/seer												
Kidney bean (lobia)		17	afghanis/seer												
Rice		[illegible]	afghanis/seer												
Wheat		[illegible]	afghanis/seer												
Production/Biological production/Animals and animal products															
Chicken		17	afghanis/seer												
Mutton, fresh		20	afghanis/seer												
District Totals				0	0	0	0	0	0		0	0			
PROVINCE TOTALS				0	0	0	0	0	0		0	0			

OBJECTIVE: Food Security
INDICATOR: Donated Emergency Commodities

Item	Origin	Amount	Units of Measurement	Dollars USAID Contribution	Dollar Transportation Cost	Beneficiaries: Salaried: Afghan Males	Afghan Fem.	Non-Afgh.	Total Salaried	Afgh. % of Total	Unsalaried	Indirect	Project Dates: Start	End	Monitoring Status/Date
BADAKSHAN PROVINCE															
Baharak District															
Relief/Commodities/Food															
Chickpea (dal chana, garbanzo)	Afghanaid	1	metric tons	1000	50										2 09/09/09
Chickpea (dal chana, garbanzo)	Afrane	10	metric tons	10000	1000										1 09/09/09
Chickpea (dal chana, garbanzo)	CARE	10	metric tons	10000	1000										1 09/09/09
Wheat	Mercy Fund	10	metric tons	10000	1000										-1 09/09/09
Wheat	Save Cldrn US	10	metric tons	10000	1000										0 09/09/09
Relief/Commodities/Operating costs															
Field salaries	Afghanaid	0	dollars	2500		25	25	25	75	67%	25	25	09/09/09	09/09/09	2 09/09/09
Field salaries	Afrane	0	dollars	2500		25	25	25	75	67%	25	25	09/09/09	09/09/00	-1 09/09/09
Field salaries	CARE	0	dollars	2500		25	25	25	75	67%	25	25	09/09/09	09/09/09	2 09/09/09
Field salaries	Mercy Fund	0	dollars	2500		25	25	25	75	67%	25	25	09/09/09	09/09/09	2 09/09/09
Field salaries	Save Cldrn US	0	dollars	25000		25	25	25	75	67%	25	25	09/09/09	09/09/09	1 09/09/09
District Totals				76000	4050	125	125	125	375	67%	125	125			
Zebak District															
Relief/Commodities/Food															
Chickpea (dal chana, garbanzo)	Afghanaid	10	metric tons	10000	1000										2 09/09/09
Wheat	Afghanaid	100	metric tons	100000	10000										2 09/09/09
Wheat	CARE	200	metric tons	200000	20000										1 09/09/09
Relief/Commodities/Operating costs															
Field salaries	Afghanaid	0	dollars	2500		25	25	25	75	67%	25	25	09/09/09	09/09/09	2 09/09/09
Field salaries	CARE	0	dollars	2500		25	25	25	75	67%	25	25	09/09/09	09/09/09	2 09/09/09
District Totals				315000	31000	50	50	50	150	67%	50	50			
PROVINCE TOTALS				[illegible]	35050	175	175	175	525	67%	175	175			
ZABUL PROVINCE															
Arghandab District															
Relief/Commodities/Food															
Wheat	Afghanaid	100	metric tons	100000	10000										2 09/09/09
Wheat	CARE	100	metric tons	10000	1000										2 09/09/09
Relief/Commodities/Operating costs															
Field salaries	Afghanaid	0	dollars	25000		25	25	25	75	67%	250	250	09/09/09	09/09/09	2 09/09/09
District Totals				135000	11000	25	25	25	75	67%	250	250			
PROVINCE TOTALS				135000	11000	25	25	25	75	67%	250	250			

USAID/AFGHANISTAN: PERFORMANCE INDICATORS FOR Q1 1991 BY OBJECTIVE

OBJECTIVE: Food Security
INDICATOR: Traded Commodities

Item	Origin	Amount	Units of Measurement	Dollars USAID Contribution	Dollar Transportation Cost	Beneficiaries: Salaried: Afghan Males	Afghan Fem.	Non-Afgh.	Total Salaried	Afgh. % of Total	Unsalaried	Indirect	Project Dates: Start	End	Monitoring Status/Date
BADAKSHAN PROVINCE															
Baharak District															
Production/Biological production/Plants and plant products															1 09/09/09
Kidney bean (lubia)	DAI	1	metric tons	5000	500										1 09/09/09
Maize	DAI	1	metric tons	2500	250										0 09/09/09
Poplar	DAI	1	metric tons	2500	250										
Production/Biological production/Chemical inputs and feed															1 09/09/09
DAP	DAI	10	metric tons	100000	11000										-1 09/09/09
Urea	DAI	1	metric tons	100000	10000										
Production/Biological production/Health care															1 09/09/09
Pesticides, plant...	DAI	1	metric tons?	250000	2500										
Production/Biological production/Operating costs															
Field salaries	DAI	0	dollars	25000		25	25	25	75	67%	25	25	09/09/09	09/09/09	1 09/09/09
District Totals				485000	24500	25	25	25	75	67%	25	25			
PROVINCE TOTALS				485000	24500	25	25	25	75	67%	25	25			
HERAT PROVINCE															
Adraskan District															
Production/Biological production/Plants and plant products															
Poplar	DAI	0	metric tons												
District Totals				0	0	0	0	0	0		0	0			
PROVINCE TOTALS				0	0	0	0	0	0		0	0			
ZABUL PROVINCE															
Arghandab District															
Production/Biological production/Plants and plant products															1 09/09/09
Kidney bean (lubia)	DAI	1	metric tons	2500	250										0 09/09/09
Maize	DAI	1	metric tons	2500	250										2 09/09/09
Poplar	DAI	1	metric tons	2500	250										
Production/Biological production/Operating costs															
Field salaries	DAI	0	dollars	2500		25	25	25	75	67%	25	25	09/09/09	09/09/09	1 09/09/09
Field salaries	DAI	0	dollars	2500		25	25	25	75	67%	25	25	09/09/09	09/09/09	2 09/09/09
District Totals				12500	750	50	50	50	150	67%	50	50			
PROVINCE TOTALS				12500	750	50	50	50	150	67%	50	50			

OBJECTIVE: Food Security

INDICATOR: Infrastructure Created or Rehabilitated

Item	Origin	Amount	Units of Measurement	Dollars USAID Contribution	Dollar Transportation Cost	Beneficiaries: Salaried: Afghan Males	Afghan Fem.	Non-Afgh.	Total Salaried	Afgh. % of Total	Unsalaried	Indirect	Project Dates: Start	End	Monitoring Status/Date
BADAKSHAN PROVINCE															
Baharak District															
Infrastructure/Structures/Environmental modeling															
Gabion	Afghanaid	20	units	200	10										0 09/09/09
Terrace	Afrane	500	square meters	200											1 09/09/09
Infrastructure/Structures/Water resource, deliv, removal															
Canal	CARE	2000	meters	5000											2 09/09/09
Canal	CHA	0	meters	8000											2 09/09/09
Canal	Solidarites	5000	meters	2000	20										2 09/09/09
Kariz	VITA	5000	meters	2000	20										1 09/09/09
Infrastructure/Structures/Buildings															
House	CARE	5	buildings	5000	1000										0 09/09/09
Infrastructure/Structures/Transportation infrastructure															
Bridge, Bailey	CCSC	1	units	10000	1000										0 09/09/09
Causeway	CCSC	1	units	500	50										0 09/09/09
Crossing, Irish (wash lining)	CCSC	1	units	1000	100										-1 09/09/09
Road, main	CCSC	50	kilometers	100000	1000										2 09/09/09
Road, rural (farm to market)	CARE	20	kilometers	40000	4000										1 09/09/09
Road, rural (farm to market)	Save Cldrn US	20	kilometers	20000	2000										1 09/09/09
Road, rural (farm to market)	Solidarites	20	kilometers	20000	2000										1 09/09/09
Road, rural (farm to market)	VITA	20	kilometers	20000	2000										1 09/09/09
Infrastructure/Structures/Raw materials; parts															
Chicken wire	DAI	[illegible]	rolls	250	[illegible]										-1 09/09/09
Roof sheeting, plastic	DAI	100	square meters	30	20										-1 09/09/09
Infrastructure/Structures/Operating costs															
Field salaries	Afghanaid	0	dollars	15000		25	25	25	75	67%	25	25	09/09/09	09/09/09	2 09/09/09
Field salaries	Afrane	0	dollars	20000		25	25	25	75	67%	25	25	09/09/09	09/09/09	2 09/09/09
Field salaries	CARE	0	dollars	20000		25	25	25	75	67%	25	25	09/09/09	09/09/09	2 09/09/09
Field salaries	CCSC	0	dollars	20000		25	25	25	75	67%	25	25	09/09/09	09/09/09	2 09/09/09
Field salaries	CHA	0	dollars	20000		25	25	25	75	67%	25	25	09/09/09	09/09/90	2 09/09/09
Field salaries	DAI	0	dollars	20000		25	25	25	75	67%	25	25	09/09/09	09/09/09	2 09/09/09
Field salaries	Hogar	0	dollars	20000		25	25	25	75	67%	25	25	09/09/09	09/09/09	2 09/09/09
Field salaries	Save Cldrn US	[illegible]	[illegible]	[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Field salaries	Solidarites	0	dollars	20000		25	25	25	75	67%	25	25	09/09/09	09/09/09	2 09/09/09
Field salaries	VITA	0	dollars	20000		25	25	25	75	67%	25	25	09/09/09	09/09/09	2 09/09/09
District Totals				429180	13240	250	250	250	750	67%	250	250			
PROVINCE TOTALS				429180	13240	250	250	250	750	67%	250	250			

OBJECTIVE: Sustainable Health/Education Services

INDICATOR: Operating Costs; Number of Afghans Providing Services

Item	Origin	Amount	Units of Measurement	Dollars USAID Contribution	Dollar Transportation Cost	Beneficiaries: Salaried: Afghan Males	Afghan Fem.	Non-Afgh.	Total Salaried	Afgh. % of Total	Unsalaried	Indirect	Project Dates: Start	End	Monitoring Status/Date
BADAKSHAN PROVINCE															
Baharak District															
Health/Preventive med. facility support/MCH posts															
Field salaries	MSH	1	facilities	200			1		1	100%	100		09/09/09		1 09/09/09
Supplies and med. equipment	MSH	0	dollars	50	10										1 09/09/09
Health/Clinical care facility support/Basic health post															
Field salaries	MSH	1	facilities	200		1			1	100%	50		09/09/09		0 09/09/09
Field salaries	MSH	1	facilities	200		1			1	100%	100		09/09/09		1 09/09/09
Pharmaceuticals	MSH	0	dollars	50	5										0 09/09/09
Pharmaceuticals	MSH	0	dollars	50											1 09/09/09
Supplies and med. equipment	MSH	0	dollars	50	10										2 09/09/09
Supplies and med. equipment	MSH	0	dollars	200	20										1 09/09/09
Health/Clinical care facility support/Primary care hospital															
Field salaries	MCI	1	facilities	2000		9		1	10	90%	1200		09/09/09		1 09/09/09
Field salaries	MSH	1	facilities	2000		8		2	10	80%	1000		09/09/09		2 09/09/09
Pharmaceuticals	MCI	0	dollars	1000	100										1 09/09/09
Pharmaceuticals	MSH	0	dollars	300	30										2 09/09/09
Supplies and med. equipment	MCI	0	dollars	1000	100										1 09/09/09
Supplies and med. equipment	MSH	0	dollars	1000	100										1 09/09/09
Education/Educational facility support/Primary schools															
Field salaries	UNO	1	facilities	200		1			1	100%	50		09/09/09		2 09/09/09
Supplies	UNO	0	dollars	20	10										1 09/09/09
Textbooks	UNO	0	dollars	40	10										1 09/09/09
Education/Educational facility support/Secondary schools															
Field salaries	UNO	1	facilities	2000		10			10	100%	1000		09/09/09		1 09/09/09
Supplies	UNO	0	dollars	500	100										1 09/09/90
Textbooks	UNO	0	dollars	1000	100										1 09/09/09
District Totals				12060	595	30	1	3	34	91%	3500	0			
PROVINCE TOTALS				12060	595	30	1	3	34	91%	3500	0			

OBJECTIVE: Reduce Poppy Production/Increase Narcotics Awareness
INDICATOR: Poppy Production in Monitored Areas

Item	Origin	Amount	Units of Measurement	Dollars USAID Contribution	Dollar Transportation Cost	Beneficiaries: Salaried: Afghan Males	Afghan Fem.	Non-Afgh.	Total Salaried	Afgh. % of Total	Unsalaried	Indirect	Project Dates: Start	End	Monitoring Status/Date
BADAKSHAN PROVINCE															
Baharak District															
Production/Biological production/Plants and plant products															
Poppy, opium	DAI	40	jeribs												
Poppy, opium	DAI	20	jeribs												
District Totals				0	0	0	0	0	0		0	0			
PROVINCE TOTALS				0	0	0	0	0	0		0	0			
HERAT PROVINCE															
Adraskan District															
Production/Biological production/Plants and plant products															
Poppy, opium	DAI	0	jeribs												
District Totals				0	0	0	0	0	0		0	0			
PROVINCE TOTALS				0	0	0	0	0	0		0	0			
ZABUL PROVINCE															
Arghandab District															
Production/Biological production/Plants and plant products															
Poppy, opium	DAI	23	jeribs												
District Totals				0	0	0	0	0	0		0	0			
PROVINCE TOTALS				0	0	0	0	0	0		0	0			

OBJECTIVE: Reduce Poppy Production/Increase Narcotics Awareness
INDICATOR: Persons Trained in Narcotics Awareness

Item	Origin	Amount	Units of Measurement	Dollars USAID Contribution	Dollar Transportation Cost	Beneficiaries: Salaried: Afghan Males	Afghan Fem.	Non-Afgh.	Total Salaried	Afgh. % of Total	Unsalaried	Indirect	Project Dates: Start	End	Monitoring Status/Date
BADAKSHAN PROVINCE															
Baharak District															
Public policy and admin./Public information/Public information campaigns															
Narcotics awareness training	DAI	20	hours								200				0
District Totals				0	0	0	0	0	0		200	0			
PROVINCE TOTALS				0	0	0	0	0	0		200	0			
ZABUL PROVINCE															
Arghandab District															
Public policy and admin./Public information/Public information campaigns															
Narcotics awareness training	DAI	10	hours								200				0
District Totals				0	0	0	0	0	0		200	0			
PROVINCE TOTALS				0	0	0	0	0	0		200	0			

OBJECTIVE: Equitable Access to AID-Supported Training
INDICATOR: Afghans Trained in Vocational Skills

Item	Origin	Amount	Units of Measurement	Dollars USAID Contribution	Dollar Transportation Cost	Beneficiaries: Salaried: Afghan Males	Afghan Fem.	Non-Afgh.	Total Salaried	Afgh. % of Total	Unsalaried	Indirect	Project Dates: Start	End	Monitoring Status/Date
BADAKSHAN PROVINCE															
Baharak District															
Infrastructure/Human resource development/Training															
Construction supervision	CCSC	35	hours	200		10			10	100%					0
Infrastructure/Human resource development/Conferences and workshops															
Support for events	VITA	0	dollars	600			30		30	100%					0
Production/Human resource development/Training and extension															
Ag. marketing agent training	DAI	35	hours	200		10			10	100%					0
Demonstration farm	DAI	1	farms	500		2			2	100%					0
Production/Human resource development/Conferences and workshops															
Support for events	Save Cldrn US	0	dollars	600			30		30	100%					0
Health/Human resource development/Training															
Female health worker (FHW/Dai)	MSH	35	hours	500			4	80	84	5%					1 09/09/09
Vaccinator	DAI	14	hours	200		10			10	100%					2 09/09/09
Education/Human resource development/Training															
Supervisory teacher training	UNO	35	hours	500		10			10	100%					0
Education/Human resource development/Conferences and workshops															
Support for events	UNO	0	dollars	600			30		30	100%			09/09/09		0
District Totals				3900	0	42	94	80	216	63%	0	0			
PROVINCE TOTALS				3900	0	42	94	80	216	63%	0	0			

OBJECTIVE: Equitable Access to AID-Supported Trainig

INDICATOR: Services Provided by Afghans in Aid-Supported Organizations

Item	Origin	Amount	Units of Measurement	Dollars USAID Contribution	Dollar Transportation Cost	Beneficiaries: Salaried: Afghan Males	Afghan Fem.	Non-Afgh.	Total Salaried	Afgh. % of Total	Unsaliried	Indirect	Project Dates: Start	End	Monitoring Status/Date

OBJECTIVE: More Open Society
INDICATOR: Events with DPI Themes

Item	Origin	Amount	Units of Measurement	Dollars USAID Contribution	Dollar Transportation Cost	Beneficiaries: Salaried: Afghan Males	Afghan Fem.	Non-Afgh.	Total Salaried	Afgh. % of Total	Unsalaried	Indirect	Project Dates: Start	End	Monitoring Status/Date
BADAKSHAN PROVINCE															
Baharak District															
Infrastructure/Human resource development/Conferences and workshops															
Support for events	VITA	0	dollars	600			30		30	100%					0
Production/Human resource development/Conferences and workshops															
Support for events	Save Cldrn US	0	dollars	600			30		30	100%					0
Education/Human resource development/Conferences and workshops															
Support for events	UNO	0	dollars	600			30		30	100%			09/09/09		0
District Totals				1800	0	0	90	0	90	100%	0	0			
PROVINCE TOTALS				1800	0	0	90	0	90	100%	0	0			

OBJECTIVE: More Open Society

INDICATOR: Afghan Participation in International Forums

Item	Origin	Amount	Units of Measurement	Dollars USAID Contribution	Dollar Transportation Cost	Beneficiaries: Salaried: Afghan Males	Afghan Fem.	Non-Afgh.	Total Salaried	Afgh. % of Total	Unsalaried	Indirect	Project Dates: Start	End	Monitoring Status/Date
REPORT TOTALS				1470440	85135	597	710	608	1915	68%	4650	750			

USAID/AFGHANISTAN: PERFORMANCE INDICATORS FOR Q1 1991 BY LOCATION

BADAKSHAN PROVINCE, BAHARAK DISTRICT

Item	Origin	Amount	Units of Measurement	Dollars USAID Contribution	Dollar Transportation Cost	Beneficiaries: Salaried: Afghan Males	Afghan Fem.	Non-Afgh.	Total	% of Total	Unsalaried	Indirect	Project Dates: Start	End	Monitoring Status: 1	0	-1	2	Date
BAHARAK DISTRICT																			
Objective: Food Security; Indicator: Market Prices for Basic Foods																			
Production/Biological production/Plants and plant products																			
Chickpea (dal chana/garbanzo)		20	afghanis/seer																
Chickpea (dal chana/garbanzo)		22	afghanis/seer																
Kidney bean (lubia)		15	afghanis/seer																
Rice		9	afghanis/seer																
Wheat		10	afghanis/seer																
Production/Biological production/Animals and animal products																			
Chicken		25	afghanis/seer																
Chicken eggs, fresh		17	afghanis/seer																
Mutton, fresh		25	afghanis/seer																
Indicator Totals				0	0	0	0	0	0		0	0			0	0	0	0	
Objective: Food Security; Indicator: Donated Emergency Commodities																			
Relief/Commodities/Food																			
Chickpea (dal chana, garbanzo)	**Afghanaid**	**1**	**metric tons**	**1000**	**50**													1	09/09/09
Chickpea (dal chana, garbanzo)	Afrane	10	metric tons	10000	1000										1				09/09/09
Chickpea (dal chana, garbanzo)	**CARE**	**10**	**metric tons**	**10000**	**1000**										1				09/09/09
Wheat	**Mercy Fund**	**10**	**metric tons**	**10000**	**1000**												1		09/09/09
Wheat	**Save Cldrn US**	**10**	**metric tons**	**10000**	**1000**														09/09/09
Relief/Commodities/Operating costs																			
Field salaries	Afghanaid	0	dollars	2500		25	25	25	75	67%	25	25	09/09/09	09/09/09				1	09/09/09
Field salaries	Afrane	0	dollars	2500		25	25	25	75	67%	25	25	09/09/09	09/09/00			1		09/09/09
Field salaries	CARE	0	dollars	2500		25	25	25	75	67%	25	25	09/09/09	09/09/09				1	09/09/09
Field salaries	Mercy Fund	0	dollars	2500		25	25	25	75	67%	25	25	09/09/09	09/09/09				1	09/09/09
Field salaries	Save Cldrn US	0	dollars	2500		25	25	25	75	67%	25	25	09/09/09	09/09/09	1				09/09/09
Indicator Totals				76000	4050	125	125	125	375	67%	125	125			3	1	2	4	
Objective: Food Security; Indicator: Traded Commodities																			
Production/Biological production/Plants and plant products																			
Kidney bean (lubia)	DAI	1	metric tons	500	50										1				09/09/09
Maize	DAI	1	metric tons	2500	250										1				09/09/09
Poplar	DAI	1	metric tons	2500	250											1			09/09/09
Production/Biological production/Chemical inputs and feed																			
DAP	DAI	10	metric tons	100000	11000										1				09/09/09
Urea	DAI	1	metric tons	100000	10000												1		09/09/09
Production/Biological production/Health care																			
Pesticides, plant...	DAI	1	metric tons?	250000	2500										1				09/09/09
Production/Biological production/Operating costs																			
Field salaries	DAI	0	dollars	25000		25	25	25	75	67%	25	25	09/09/09	09/09/09	1				09/09/09
Indicator Totals				485000	24500	25	25	25	75	67%	25	25			5	1	1	0	
Objective: Food Security; Indicator: Infrastructure Created or Rehabilitated																			
Infrastructure/Structures/Environmental modeling																			
Gabion	**Afghanaid**	**20**	**units**	**200**	**10**											1			09/09/09

BADAKSHAN PROVINCE, BAHARAK DISTRICT

Item	Origin	Amount	Units of Measurement	Dollars USAID Contri-bution	Dollar Trans-porta-tation Cost	Beneficiaries: Salaried: Afghan Males	Afghan Fem.	Non-Afgh.	Total	Afgh. % of Total	Unsal-aried	Indi-rect	Project Dates: Start	End	Monitoring Status: 1	0	-1	2	Date
Terrace	Afrane	500	square meters	200											1				09/09/09
Infrastructure/Structures/Water resource, deliv, removal																			
Canal	CARE	2000	meters	5000														1	09/09/09
Canal	CHA	0	meters	8000														1	09/09/09
Canal	Solidarites	5000	meters	2000	20													1	09/09/09
Kariz	VITA	5000	meters	2000	20										1				09/09/09
Infrastructure/Structures/Buildings																			
House	CARE	5	buildings	5000	1000											1			09/09/09
Infrastructure/Structures/Transportation infrastructure																			
Bridge, Bailey	CCSC	1	units	10000	1000											1			09/09/09
Causeway	CCSC	1	units	500	50											1			09/09/09
Crossing, Irish (wash lining)	CCSC	1	units	1000	100												1		09/09/09
Road, main	CCSC	50	kilometers	100000	1000													1	09/09/09
Road, rural (farm to market)	CARE	20	kilometers	40000	4000										1				09/09/09
Road, rural (farm to market)	Save Cldrn US	20	kilometers	20000	2000										1				09/09/09
Road, rural (farm to market)	Solidarites	20	kilometers	20000	2000										1				09/09/09
Road, rural (farm to market)	VITA	20	kilometers	20000	2000										1				09/09/09
Infrastructure/Structures/Raw materials; parts																			
Chicken wire	DAI	2	rolls	250	20												1		09/09/09
Roof sheeting, plastic	DAI	100	square meters	30	20												1		09/09/09
Infrastructure/Structures/Operating costs																			
Field salaries	Afghanaid	0	dollars	15000		25	25	25	75	67%	25	25	09/09/09	09/09/09				1	09/09/09
Field salaries	Afrane	0	dollars	20000		25	25	25	75	67%	25	25	09/09/09	09/09/09				1	09/09/09
Field salaries	CARE	0	dollars	20000		25	25	25	75	67%	25	25	09/09/09	09/09/09				1	09/09/09
Field salaries	CCSC	0	dollars	20000		25	25	25	75	67%	25	25	09/09/09	09/09/09				1	09/09/09
Field salaries	CHA	0	dollars	20000		25	25	25	75	67%	25	25	09/09/09	09/09/90				1	09/09/09
Field salaries	DAI	0	dollars	20000		25	25	25	75	67%	25	25	09/09/09	09/09/09				1	09/09/09
Field salaries	Hodar	0	dollars	20000		25	25	25	75	67%	25	25	09/09/09	09/09/09				1	09/09/09
Field salaries	Save Cldrn US	0	dollars	20000		25	25	25	75	67%	25	25	09/09/09	09/09/09				1	09/09/09
Field salaries	Solidarites	0	dollars	20000		25	25	25	75	67%	25	25	09/09/09	09/09/09				1	09/09/09
Field salaries	VITA	0	dollars	20000		25	25	25	75	67%	25	25	09/09/09	09/09/09				1	09/09/09
Indicator Totals				429180	13240	250	250	250	750	67%	250	250			6	4	3	14	
Objective: Sustainable Health Education Services. Indicator: Indicator: Number of Afghans [illegible] services																			
Health/Preventive med. facility support/MCH posts (1 facility)																			
Field salaries	MSH	1	facilities	200			1		1	100%	100		09/09/09		1				09/09/09
Supplies and med. equipment	MSH	0	dollars	50	10										1				09/09/09
Health/Clinical care facility support/Basic health post (2 facilities)																			
Field salaries	MSH	1	facilities	200		1			1	100%	50		09/09/09			1			09/09/09
Field salaries	MSH	1	facilities	200		1			1	100%	100		09/09/09		1				09/09/09
Pharmaceuticals	MSH	0	dollars	50	5											1			09/09/09
Pharmaceuticals	MSH	0	dollars	50											1				09/09/09
Supplies and med. equipment	MSH	0	dollars	50	10													1	09/09/09
Supplies and med. equipment	MSH	0	dollars	200	20										1				09/09/09
Health/Clinical care facility support/Primary care hospital (2 facilities)																			
Field salaries	MCI	1	facilities	2000		9		1	10	90%	1200		09/09/09		1				09/09/09
Field salaries	MSH	1	facilities	2000		8		2	10	80%	1000		09/09/09					1	09/09/09

BADAKSHAN PROVINCE, BAHARAK DISTRICT

Item	Origin	Amount	Units of Measurement	Dollars USAID Contribution	Dollar Transportation Cost	Beneficiaries: Salaried: Afghan Males	Afghan Fem.	Non-Afgh.	Total	Afgh. % of Total	Unsalaried	Indirect	Project Dates: Start	End	Monitoring Status: 1	0	-1	2	Date
Pharmaceuticals	MCI	0	dollars	1000	100										1				09/09/09
Pharmaceuticals	MSH	0	dollars	300	30													1	09/09/09
Supplies and med. equipment	MCI	0	dollars	1000	100										1				09/09/09
Supplies and med. equipment	MSH	0	dollars	1000	100										1				09/09/09
Education/Educational facility support/Primary schools (1 facility)																			
Field salaries	UNO	1	facilities	200		1			1	100%	50		09/09/09					1	09/09/09
Supplies	UNO	0	dollars	20	10										1				09/09/09
Textbooks	UNO	0	dollars	40	10										1				09/09/09
Education/Educational facility support/Secondary schools (1 facility)																			
Field salaries	UNO	1	facilities	2000		10			10	100%	1000		09/09/09		1				09/09/09
Supplies	UNO	0	dollars	500	100										1				09/09/90
Textbooks	UNO	0	dollars	1000	100										1				09/09/09
Indicator Totals				12060	595	30	1	3	34	91%	3500	0			14	2	0	4	
Objective: Reduce Poppy Production/Increase Narcotics Awareness; Indicator: Poppy Production in Monitored Areas																			
Production/Biological production/Plants and plant products																			
Poppy, opium	DAI	40	jeribs																
Poppy, opium	DAI	20	jeribs																
Indicator Totals				0	0	0	0	0	0		0	0			0	0	0	0	
Objective: Reduce Poppy Production/Increase Narcotics Awareness; Indicator: Persons Trained in Narcotics Awareness																			
Public policy and admin./Public information/Public information campaigns																			
Narcotics awareness training	DAI	20	hours								200					1			
Indicator Totals				0	0	0	0	0	0		200	0			0	1	-	1	
Objective: Equitable Access to AID-Supported Training; Indicator: Afghans Trained in Vocational Skills																			
Infrastructure/Human resource development/Training																			
Construction supervision	CCSC	35	hours	200		10			10	100%						1			
Infrastructure/Human resource development/Conferences and workshops																			
Support for events	VITA	0	dollars	600			30		30	100						1			
Production/Human resource development/Training and extension																			
Ag. marketing agent training	DAI	35	hours	200		10			10	100%						1			
Demonstration farm	DAI	1	farms	500		1			1	100%						1			
Production/Human resource development/Conferences and workshops																			
Support for events	Save Cldrn US	0	dollars	600			30		30	100%						1			
Health/Human resource development/Training																			
Female health worker (FHW/Dai)	MSH	35	hours	500			4	80	84	5%					1				09/09/09
Vaccinator	DAI	14	hours	200		10			10	100%								1	09/09/09
Education/Human resource development/Training																			
Supervisory teacher training	UNO	35	hours	500		10			10	100%						1			
Education/Human resource development/Conferences and workshops																			
Support for events	UNO	0	dollars	600			30		30	100%			09/09/09			1			
Indicator Totals				3900	0	42	94	80	216	63%	0	0			1	7	0	1	

BADAKSHAN PROVINCE, BAHARAK DISTRICT

Item	Origin	Amount	Units of Measurement	Dollars USAID Contribution	Dollar Transportation Cost	Beneficiaries: Salaried Afghan Males	Salaried Afghan Fem.	Salaried Non-Afgh.	Salaried Total	Afgh. % of Total	Unsalaried	Indirect	Project Dates Start	Project Dates End	Monitoring Status 1	0	-1	2	Date
Objective: More Open Society; Indicator: Events with DPI Themes																			
Infrastructure/Human resource development/Conferences and workshops																			
Support for events	VITA	0	dollars	600			30		30	100%						1			
Production/Human resource development/Conferences and workshops																			
Support for events	Save Cldrn US	0	dollars	600			30		30	100%						1			
Education/Human resource development/Conferences and workshops																			
Support for events	UNO	0	dollars	600			30		30	100%			09/09/09			1			
Indicator Totals				1800	0	0	90	0	90	100%	0	0			0	3	0	0	
DISTRICT TOTALS				1007940	42385	472	585	483	1540	69%	4100	400			29	19	6	23	
PROVINCE TOTALS				1007940	42385	472	585	483	1540	69%	4100	400			29	19	6	23	

ALA.DBF

ALA	Identifying code for first level activity.
ALANAME	Meaning of code ALA.

ALB.DBF

ALA	Identifying code for first level activity.
ALB	Identifying code for second level activity.
ALBNAME	Meaning of code for ALA+ALB+ALC.

ALC.DBF

ALA	Identifying code for first level activity.
ALB	Identifying code for second level activity.
ALC	Identifying code for third level activity.
ALCNAME	Meaning of code for ALA+ALB+ALC.

ALD.DBF

ALA	Identifying code for first level activity.
ALB	Identifying code for second level activity.
ALC	Identifying code for third level activity.
ALD	Identifying code for fourth level activity
ALDNAME	Meaning of code for ALA+ALB+ALC+ALD.
MEAS	Units of measurement for the item.
PRICEUNIT	Price per unit for the item.
INDIC	List of codes for indictors to which the item pertains.
VWDPI	Whether this item should display a prompt for DPI.
VWMEAS	Whether this item should display the measure.
VWAMT	Whether this item should display an amount
VWTRNSP	Whether this item should display tranportation costs
VWAFGHS	Whether this item should ask for Afghan salaried beneficiaries
VWNAFGHS	Whetheer this item should ask for nonAfghan salaried beneficiaries.
VWUNSAL	Whether this item should ask for unsalaried beneficiaries.
VWINDIR	Whether this item should ask for indirect beneficiaries
VWDATES	Whether this item should ask for activity dates
VWVERIF	Whether this item should ask for monitoring information.

CONTRACT

CONNO	Contractor code.
CONABBREV	Abbreviated contractor name for printing on reports
CONNAME	Full contractor name.

DISTRICT.DBF

PROVNO	Province code.
DISTNO	District code within province
DISTRICT	Name of district.

INDIC

INDNO	Indicator number
OBJECTIVE	Objective to which the indicator pertains
INDICATOR	Name of indicator
COMMENT	Comment line for use in editing indicators

NEWPLACE.DBF

PROVNO	Province code.
DISTNO	District code within province
PLACE	Name of new village not listed in database
MENTIONED	Field for future use, where the number of instances of a new village having been entered can be totalled.

PARTY

PARTYNO	Party code
PARTY	Party name

PLACE.DBF

PROVNO	Province code.
DISTNO	District code within province.
PLACENO	Village or city code within district.
PLACE	Name of village.

PROVINCE.DBF

PROVNO	Province code.
PROVINCE	Name of province.

REPDATA.DBF

QTR	Quarter to which the record pertains
YEAR	Year to which the record pertains
PROVNO	Province code
DISTNO	District code within province
PLACENO	Village code within district
ALA	First-level activity code
ALB	Second-level activity code
ALC	Third-level activity code
ALD	Fourth-level activity code
CONNO	Contractor code
CONABBREV	Abbreviated contractor name
SEQ	Activity sequence
SUBGRANT	Subgrant number
QTY	Quantity of the item being measured
AIDAMT	Dollar value of USAID contribution
TRANSPORT	Transportation cost of inputs
MAFSBENE	Male Afghan salaried beneficiaries
FAFSBENE	Female Afghan salaried beneficiaries
NAFSBENE	Non-Afghan salaried beneficiaries
UNSBENE	Unsalaried beneficiaries
INDBENE	Indirect beneficiaries
DPI	Whether the record supports DPI
DONATED	Whether the input was donated (T) or traded (F)
PRICE	Whether the record is one of a price
BEGIN	Beginning date for activity
END	Ending date for activity
VERSTAT	Verification status
VERDATE	Verification date
PARTYNO	Party code
INTERMED	Name of Intermediary/commander/shura
UPDATE	Last date the record was changed

CONTENTS

Module	File	Source Listing	Local Concordance	Module Cross-reference
CHPARTY	CHPARTY.PRG	59	61	85
CONT	CONT.PRG	56	57	85
DEFAULT	DEFAULT.PRG	17	18	85
ENTRY	ENTRY.PRG	3	14	86
LOCATION	LOCATION.PRG	38	45	86
REBUILD	REBUILD.PRG	64	64	86
REP	REP.PRG	1	1	87
RPTLOC	RPTLOC.PRG	27	33	87
RPTLOCHD	RPTLOCHD.PRG	63	63	87
RPTOBJ	RPTOBJ.PRG	19	24	88
RPTOBJHD	RPTOBJHD.PRG	62	62	88
SECTOR	SECTOR.PRG	46	54	88
UTIL	UTIL.PRG	36	37	89

Global Concordance 66

Hierarchical Tree 84

DBF's

ALA	70
ALB	71
ALC	72
ALD	73
CONTRACT	75
DISTRICT	76
INDIC	77
NEWPLACE	78
PARTY	79
PLACE	80
PROVINCE	81
REPDATA	82

```
 1 * rep.prg - menus for O/AID/Rep Data Collection & Analysis program
 2 * ver 1 Ron Barney 21 October 1990
 3
 4 set stat off
 5 set bell off
 6 set talk off
 7
 8 ** global default variables stored in pantry.mem:
 9 ** dfconno, dfconname, dfqtr, dfyear, dfprice, dfsubgrant
10 restore from pantry.mem
11
12 DO WHILE .T.
13 W    clear
14 W    @  1,20 say 'USAID/Representative for Afghanistan/DC&A'
15 W    @  2,29 say 'PERFORMANCE DATA SYSTEM'
16 W    @  3,1 to 3,78
17 W    @  7,24 say '1 - Enter/Modify/Delete Records'
18 W    @  9,24 say '2 - Set Defaults'
19 W    @ 11,24 say '3 - Print Reports'
20 W    @ 13,24 say '4 - Perform Utilities'
21 W    answer='0'
22 W    @ 16,24 say 'Selection: ' get answer pict '!'
23 W    @ 22,1 say  'PROTOTYPE 11/25/90'
24 W    read
25 W    DO CASE
26 W    CASE answer='0'
27 <=========EXIT && **********
28 W    CASE answer='1'
29 W   2    DO ENTRY
30 W    CASE answer='2' && set defaults
31 W   3    DO DEFAULT
32 W    CASE answer='3' && Print reports
33 W   4    DO WHILE .T.
34 W   4   W    clear
35 W   4   W    @  1,20 say 'USAID/Representative for Afghanistan/DC&A'
36 W   4   W    @  2,29 say 'PERFORMANCE DATA SYSTEM'
37 W   4   W    @  3,1 to 3,78
38 W   4   W    @  6,20 say 'Reports:'
39 W   4   W    @  8,20 say '1 - Semiannual Indicators by Objective'
40 W   4   W    @ 10,20 say '2 - Semiannual Indicators by Location'
41 W   4   W    answer='0'
42 W   4   W    @ 13,20 say 'Selection: ' get answer pict '!'
43 W   4   W    read
44 W   4   W    DO CASE
45 W   4   W    CASE answer='0'
46 W   4<===========EXIT    && from current loop back to main menu
47 W   4   W    CASE answer='1'
48 W   4   W   2    DO RPTOBJ
49 W   4   W    CASE answer='2'
50 W   4   W   3    DO RPTLOC
51 W   4   W    OTHERWISE
52 W   4   W   0    ? chr(7)
53 W   4   W    ENDCASE
54 W   4    ENDDO
55 W    CASE answer='4' && Perform utilities
56 W   5    DO UTIL
57 W    OTHERWISE       && do nothing: just loop to top
58 W    ENDCASE
59 ENDDO
60
```

++ End of Program REP ++ DOC found no errors.

VARIABLES: LOCAL CONCORDANCE FOR REP

Variable	Statement number in listing (modified variables indicated by '='										
answer:	=21	=22	26	28	30	32	=41	=42	45	47	49
	55										
off:	4	5	6								

```
1 * entry.prg - choose location, sector, contractor (and party) codes* get
1 *  rest of a single repdata.dbf record.
2 * ver 1.0 Ron Barney 23 October 1990
3 * Clipper function lastkey() has been used to test for the ESC key
4 * (ASCII 27) both here and in subprograms.  IF ESC is read from the
5 * keyboard buffer, the RETURN command is always issued.  While it is
6 * normally better structured programming to handle this kind of thing
7 * differently, there are are so many reads involved that this exceptional
8 * handling seems to make the code more clear, rather than the reverse.
9
10 ******
11 * environment:
12 * set step off
13 * set echo off
14 * set talk off
15 * set stat off
16 * set bell off
17 * set confirm off
18 * restore from pantry.mem
19 *******
20
21 select 1
22 use province index provno
23 select 2
24 use district index pdistrict, pdistno
25 select 3
26 use place index pdplace, pdplaceno
27 select 4
28 use newplace
29 select 5
30 use ALA index ALA
31 select 6
32 use ALB index ALB        && key ALA+ALB
33 select 7
34 use ALC index ALC && key ALA+ALB+ALC
35 select 8
36 use ALD index ALDname, ALD      && keys ALA+ALB+ALC+ALDname, ALA+ALB+ALC+
36 ALD
37 select 9
38 use contract index conname, conno
39 set order to 2
40 seek dfconno    && position record at default
41 vconabbrev=conabbrev
42 set order to 1
43 select 10
44 ** key for repdata.ndx: qtr+year+provno+distno+placeno+ala+alb+alc+ald+
44 * orgid+subgrant+seq
45 use repdata index repdata
46 select 11
47 use party index party, partyno
48
49 ** naming convention prefixes:  n=character data number code for which
49 *  there
50 ** is a text equivalent; c=text equivalent of n-prefix variables**
50 *  m=matching copy for another variable; v=ordinary variable copy
51 ** of a field. vw=logical field in ALD.DBF determining whether a portion
52 ** repdata.dbf is displayed or not.
53
54 ** codes for record identification:
55 store dfqtr to vqtr, mqtr        && dfqtr is global default
56 store dfyear to vyear, myear     && dfyear is global default
```

```
 57 store space(2) to        nprov, mprov
 58 store space(4) to        ndist, mdist
 59 store space(3) to nplace, mplace
 60 store space(1) to        nala,nalb,nalc, mala,malb,malc
 61 store space(2) to nald,mald
 62 store dfconno to ncon, mcon     && dfconno is global default
 63 store space(2) to        vseq, mseq
 64 store space(2) to vsubgrant, msubgrant
 65 ** labels for record-identifying codes:
 66 cprov=space(25) && label for province
 67 cdist=space(19) && label for district
 68 cplace=space(40)         && label for place
 69 store space(31) to cala,calb,calc,cald  && code character description
 70 ccon=dfconname  && global default
 71 * other variables:
 72 store .F. to vdpi,vdonated,vprice
 73 store 0 to vqty,vaidamt,vtransport,vmafsbene,vfafsbene,vnafsbene,;
 73  vuusbene,vindbene
 74 store ctod('  /  /  ') to vbegin,vend,vverdate
 75 store 0 to vverstat
 76 store space(2) to vpartyno
 77 store space(42) to vparty
 78 store space(30) to vintermed
 79 store date() to vupdate
 80 ccs=22  && column where < will appear.  Passed to programs called.
 81 cct=ccs+2        && column where text after < will begin
 82
 83 ** CHOOSE WHETHER TO ENTER, MODIFY OR DELETE
 84 DO WHILE .T.
 85 W    @ 4,0 clear
 86 W    @ 6,26 say 'Enter/Modify/Delete Records'
 87 W    @  8,26 say '1 - Enter Records'
 88 W    @ 10,26 say '2 - Modify Records'
 89 W    @ 12,26 say '3 - Delete Records'
 90 W    answer='0'
 91 W    @ 15,26 say 'Selection:' get answer pict '!'
 92 W    read
 93 W    DO CASE
 94 W    CASE answer='0' .or. lastkey()=27
 95 <---------return
 96 W    CASE answer='1'
 97 W   2    process='E'
 98 <========EXIT
 99 W    CASE answer='2'
100 W   3    process='M'
101 <========EXIT
102 W    CASE answer='3'
103 W   4    process='D'
104 <========EXIT
105 W    OTHERWISE
106 W   0    answer='0'
107 W    ENDCASE
108 ENDDO
109
110 ** DISPLAY FIRST PART OF SCREEN **************
111 * select repdata
112 clear
113 text
114 ZDDDDDDDDDDDDDDDDDDDDD
115 3       Province
116 3       District
```

```
117 3        Village
118 3
119 3 Activity level A
120 3 Activity level B
121 3 Activity level C
122 3 Activity level D
123 3
124 3
125 3       Contractor
126 3         Sequence
127 @DDDDDDDDDDDDDDDDDDDD
128 endtext
129 @ 2,ccs-3 get nprov
130 @ 3,ccs-5 get ndist
131 @ 4,ccs-4 get nplace
132 @ 6,ccs-2 get nala
133 @ 7,ccs-2 get nalb
134 @ 8,ccs-2 get nalc
135 @ 9,ccs-3 get nald
136 IF .not. dfprice
137 T    @ 12,ccs-3 get ncon
138 ELSE
139 F    vprice=.T.
140 ENDIF
141 clear gets
142
143 ** MAIN LOOP: ******************************************
144 firstpass=.T.   && first pass through screen.  Becomes .F. if screen;
144  changed
145 packdb=.F.      && controls pack command after loop
146 DO WHILE .T.
147 W    @ 15,0 clear to 24,79
148 W
149 W    DO LOCATION     && determines province, district and place code
150 W    IF lastkey()=27
151 <--------return
152 W    ENDIF
153 W
154 W    DO SECTOR       && determines activity code
155 W    IF lastkey()=27
156 <--------return
157 W    ENDIF
158 W
159 W    ** ALLOW USER TO CHANGE DFPRICE DEFAULT IF DFPRICE .T.; ELSE EXIT
160 W    IF dfprice.and.len(trim(ald->priceunit))=0
161 W   T    ? chr(7)
162 W   T    @ 10,ccs clear to 24,79
163 W   T    answer='Y'
164 W   T    @ 11,24 say 'Sorry, the system is set to collect market prices,'
165 W   T    @ 12,24 say 'not units of measurement.  Do you want to change'
166 W   T    @ 13,24 say 'the default, so that processing of this record'
167 W   T    @ 14,24 say 'can continue (Y/N)? ' get answer pict 'Y'
168 W   T    read
169 W   T    IF answer='Y'
170 W   T   T    vprice=.F.
171 W   T   T    dfprice=.F.
172 W   T   T    set safety off
173 W   T   T    save all like df* to pantry.mem
174 W   T   T    set safety on
175 W   T    ELSE
176 <------------return
```

```
177 W    T     ENDIF
178 W     ENDIF
179 W
180 W     ** FURTHER CODING IF NOT DFPRICE *******
181 W     @ 0,ccs clear to 24,79
182 W     @ 2,ccs say '< '+cprov
183 W     @ 3,ccs say '< '+cdist
184 W     @ 4,ccs say '< '+cplace
185 W     @ 6,ccs say '< '+cala
186 W     @ 7,ccs say '< '+calb
187 W     @ 8,ccs say '< '+calc
188 W     @ 9,ccs say '< '+cald
189 W     IF .not. dfprice
190 W    T     IF ald->vwdpi
191 W    T    T     @ 10,2 say "Supports DPI?" get vdpi pict 'Y'
192 W    T    T     read
193 W    T     ENDIF
194 W    T     IF lastkey()=27
195 <-------------return
196 W    T     ENDIF
197 W    T
198 W    T     IF dfconno=space(2)
199 W    T    T     DO CONT
200 W    T     ENDIF
201 W    T     IF lastkey()=27
202 <-------------return
203 W    T     ENDIF
204 W    T     @  0,ccs clear to 24,79
205 W    T     @  2,ccs say '< '+cprov
206 W    T     @  3,ccs say '< '+cdist
207 W    T     @  4,ccs say '< '+cplace
208 W    T     @  6,ccs say '< '+cala
209 W    T     @  7,ccs say '< '+calb
210 W    T     @  8,ccs say '< '+calc
211 W    T     @  9,ccs say '< '+cald
212 W    T     @ 12,ccs-3 say ncon+' < '+ccon
213 W    T     IF dfsubgrant
214 W    T    T     @ 13,24  say "Subgrant"
215 W    T    T     @ 13,33  get  vsubgrant
216 W    T    T     read
217 W    T    T     IF lastkey()=27
218 <-----------------return
219 W    T    T     ENDIF
220 W    T     ENDIF   && dfsubgrant
221 W     ENDIF && .not. dfprice
222 W
223 W     ** DISPLAY BOX, SAY/GET QUARTER, YEAR
224 W     @  1,  0  TO 14, 79
225 W     DO CASE
226 W     CASE lastkey()=27
227 <---------return
228 W     CASE vqtr<>' '.and.vyear<>'  '
229 W    2     @ 2,57 say 'Activity for Q'+vqtr+' 19'+vyear
230 W     CASE vyear<>'  '
231 W    3     @ 2,57 say 'Activity for Q  19'+vyear
232 W    3     @ 2,71 get vqtr
233 W    3     read
234 W     CASE vqtr<>' '
235 W    4     @ 2,57 say 'Activity for Q'+vqtr+' 19'
236 W    4     @ 2,75 get vyear
237 W    4     read
```

```
238 W     OTHERWISE
239 W    O     @ 2,57 say 'Activity for Q  19'
240 W    O     @ 2,71 get vqtr
241 W    O     @ 2,75 get vyear
242 W    O     read
243 W     ENDCASE
244 W     IF lastkey()=27
245 <---------return
246 W     ENDIF
247 W
248 W     ** SEQUENCE: *******
249 W     IF .not. dfprice
250 W    T     select repdata
251 W    T     IF process='E'  && record entry:  assign vseq
252 W    T    T     seek vqtr+vyear+nprov+ndist+nplace+nala+nalb+nalc+nald+ncon+
252 W    T    T     vsubgrant
253 W    T    T     IF found()       && find sequence number of the last record;
253 W    T    T      entered; add one
254 W    T    T    T     DO WHILE qtr+year+provno+distno+placeno+ala+alb+alc+ald+;
254 W    T    T    T     conno+subgrant =vqtr+vyear+nprov+ndist+nplace+nala+nalb+;
254 W    T    T    T     nalc+nald+ncon+vsubgrant
255 W    T    T    T   W     skip
256 W    T    T    T     ENDDO
257 W    T    T    T     skip-1
258 W    T    T    T     vseq=subst(str(val(seq)+101,3),2)
259 W    T    T     ELSE
260 W    T    T    F     vseq='01'
261 W    T    T     ENDIF
262 W    T    T     @ 13,19 say vseq
263 W    T     ELSE     && get existing sequence for record modify or delete
264 W    T    F     vseq=space(2)
265 W    T    F     @ 13,19 get vseq
266 W    T    F     read
267 W    T    F     IF lastkey()=27
268 <-----------------return
269 W    T    F     ENDIF
270 W    T    F     vseq=subst(str(val(vseq)+100,3),2)       && add leading zero
271 W    T     ENDIF
272 W     ENDIF    && .not. dfprice         ** end of sequence assignment
273 W
274 W     ** ASSIGN VARIABLES FOR LOWER PART OF SCREEN:***************
275 W     DO CASE
276 W     CASE process='E'
277 W    1     IF firstpass     && else do nothing, leaving variables alone
278 W    1    T     store .F. to vdpi, vdonated
279 W    1    T     store 0 to vqty,vaidamt,vtransport,vmafsbene,vfafsbene,;
279 W    1    T     vnafsbene, vunsbene,vindbene
280 W    1    T     store ctod('  /  /  ') to vbegin,vend,vverdate
281 W    1    T     store 0 to vverstat
282 W    1    T     store space(2) to vpartyno
283 W    1    T     store space(42) to vparty
284 W    1    T     store space(30) to vintermed
285 W    1    T     store date() to vupdate
286 W    1     ENDIF
287 W    1     * case 2nd time through modify and identifying signature
287 W    1     *  unchanged:
288 W     CASE process='M' .and..not. firstpass .and.;
288 W      (vqtr=mqtr.and.vyear=myear;
288 W      .and.nprov=mprov.and.ndist=mdist.and.nplace=mplace.and.nala=mala,
288 W      .and.nalb=malb.and.nalc=malc.and.nald=mald.and.ncon=mcon.and.;
288 W      vsubgrant=msubgrant.and.vseq=mseq)
```

```
289 W   2     * do nothing, leaving variables alone
290 W    OTHERWISE && process=modify or delete
291 W   O     select repdata
292 W   O     seek vqtr+vyear+nprov+ndist+nplace+nala+nalb+nalc+nald+ncon+;
292 W   O      vsubgrant+vseq
293 W   O     IF firstpass    && in deletion, firstpass always .T.
294 W   O   T     IF found()      && capture variable values from record
295 W   O   T   T     vdpi=dpi
296 W   O   T   T     vdonated=donated
297 W   O   T   T     vqty=qty
298 W   O   T   T     vaidamt=aidamt
299 W   O   T   T     vtransport=transport
300 W   O   T   T     vmafsbene=mafsbene
301 W   O   T   T     vfafsbene=fafsbene
302 W   O   T   T     vnafsbene=nafsbene
303 W   O   T   T     vunsbene=unsbene
304 W   O   T   T     vindbene=indbene
305 W   O   T   T     vbegin=begin
306 W   O   T   T     vend=end
307 W   O   T   T     vverdate=verdate
308 W   O   T   T     vverstat=verstat
309 W   O   T   T     vintermed=intermed
310 W   O   T   T     vupdate=update
311 W   O   T   T     vpartyno=partyno
312 W   O   T   T     select party
313 W   O   T   T     set order to 2  && partyno
314 W   O   T   T     seek partyno
315 W   O   T   T     vparty=party
316 W   O   T     ELSE    && .not. found()
317 W   O   T   F     ? chr(7)
318 W   O   T   F     @ 17,0
319 W   O   T   F     wait space(15)+'Record not found.  Press any key to try';
319 W   O   T   F     + ' again...'
320 ^================loop
321 W   O   T     ENDIF   && found()
322 W   O     ELSE    && .not. firstpass
323 W   O   F     IF found()
324 W   O   F   T     ? chr(7)
325 W   O   F   T     @ 18,20 say 'A record with these codes already exists.'
326 W   O   F   T     wait space(19)+'Press any key to try again...'
327 W   O   F     ELSE    && change the signature of the existing record
328 W   O   F   F     seek mqtr+myear+mprov+mdist+mplace+mala+malb+malc+mald+;
328 W   O   F   F     mcon+ msubgrant+mseq  && seek the record already found
329 W   O   F   F
330 W   O   F   F     IF .not. found()
331 W   O   F   F   T     times=0
332 W   O   F   F   T     ? chr(7)
333 W   O   F   F   T     DO WHILE times<50
334 W   O   F   F   T   W     times=times+1
335 W   O   F   F   T     ENDDO
336 W   O   F   F   T     ? chr(7)
337 W   O   F   F     ENDIF
338 W   O   F   F
339 W   O   F   F     * use same remaining variables; replace with modified
339 W   O   F   F     *  signature
340 W   O   F     ENDIF
341 W   O     ENDIF && firstpass
342 W    ENDCASE && end assignment of variables for lower part of screen
343 W
344 W    *** display rest of screen ***********************
345 W    @  3, 57  say 'Last updated '+dtoc(vupdate)
```

```
346 W     @ 15, 35  say "Male Afghan salaried beneficiaries"
347 W     @ 16, 35  say "Female Afghan salaried beneficiaries"
348 W     @ 17,  2  say "USAID contribution $"
349 W     @ 17, 35  say "Non-Afghan salaried beneficiaries"
350 W     @ 18,  2  say "Transportation"
351 W     @ 18, 35  say "Unsalaried beneficiaries"
352 W     @ 19, 35  say "Indirect beneficiaries"
353 W     @ 20,  2  say "Activity start"
354 W     @ 21,  2  say "Activity end"
355 W     @ 21, 35  say "Intermediary"
356 W     @ 22,  2  say "Status"
357 W     @ 22, 11  say ", verified"
358 W     @ 22, 35  say "Party"
359 W
360 W     @ 15,  2  get  vqty pict '9999'
361 W     IF dfprice
362 W    T    @ 15,7 say ald->priceunit
363 W     ELSE
364 W    F    @ 15,7 say ald->meas
365 W    F    ** DISFLAY gets first.  THEN go back and, if not delete,  get
365 W    F    *  them one at a
366 W    F    ** time. This allows use of lastkey() to test for escape key,
366 W    F    *  allow return.
367 W    F    IF 'B'$(ald->indic).or.'C'$(ald->indic)
368 W    F   T    @ 13, 56  say 'Donated?' get vdonated pict 'Y'
369 W    F    ENDIF
370 W    F    IF ALD->vwamt
371 W    F   T    @ 17, 23  get  vaidamt pict '99999999'
372 W    F    ENDIF
373 W    F    IF ald->vwtrnsp
374 W    F   T    @ 18, 23  get  vtransport pict '99999999'
375 W    F    ENDIF
376 W    F    IF ald->vwdates
377 W    F   T    @ 20, 23  get  vbegin
378 W    F   T    @ 21, 23  get  vend
379 W    F    ENDIF
380 W    F    IF ald->vwverif
381 W    F   T    @ 22,  9  get  vverstat pict '99'
382 W    F   T    @ 22, 23  get  vverdate
383 W    F    ENDIF
384 W    F    IF ald->vwafghs
385 W    F   T    @ 15, 72  get  vmafsbene pict '999999'
386 W    F   T    @ 16, 72  get  vfafsbene pict '999999'
387 W    F    ENDIF
388 W    F    IF ald->vwnafghs
389 W    F   T    @ 17, 72  get  vnafsbene pict '999999'
390 W    F    ENDIF
391 W    F    IF ald->vwunsal
392 W    F   T    @ 18, 70  get  vunsbene  pict '99999999'
393 W    F    ENDIF
394 W    F    IF ald->vwindir
395 W    F   T    @ 19, 70  get  vindbene  pict '99999999'
396 W    F    ENDIF
397 W    F    IF ald->vwverif
398 W    F   T    @ 21, 48  get  vintermed
399 W    F    ENDIF
400 W    F    IF ALD->vwverif
401 W    F   T    @ 22,41 get vpartyno
402 W    F    ENDIF
403 W     ENDIF   && dfprice
404 W     clear gets
```

46

```
405 W
406 W     IF process='D'  && delete
407 W   T    answer='N'
408 W   T    @ 24,29 say 'Delete this record?' get answer pict 'Y'
409 W   T    read
410 W   T    IF answer='Y'
411 W   T   T    set color to *w/n
412 W   T   T    answer='N'
413 W   T   T    @ 24,0 clear
414 W   T   T    @ 24,32 say 'Are you sure?' get answer pict 'Y'
415 W   T   T    read
416 W   T   T    set color to w/n
417 W   T   T    IF answer='Y'
418 W   T   T   T    select repdata
419 W   T   T   T    delete
420 W   T   T   T    packdb=.T.
421 W   T   T   T    answer='N'
422 W   T   T   T    @ 24,0 clear
423 W   T   T   T    @ 24,12 say  'Record has been marked for deletion. ';
423 W   T   T   T    + ' Delete another?' get answer pict 'Y'
424 W   T   T   T    read
425 W   T   T    ELSE
426 W   T   T   F    answer='N'
427 W   T   T   F    @ 24,0 clear
428 W   T   T   F    @ 24,13 say 'Record not deleted.  Delete something else'
428 W   T   T   F    + ' instead?' get answer pict 'Y'
429 W   T   T   F    read
430 W   T   T    ENDIF
431 W   T    ELSE    && answer to first request to delete='N'
432 W   T   F    answer='N'
433 W   T   F    @ 24,0 clear
434 W   T   F    @ 24,13 say 'Record not deleted.  Delete something else';
434 W   T   F    + ' instead?' get answer pict 'Y'
435 W   T   F    read
436 W   T    ENDIF
437 W   T    IF answer='Y'   && delete another or something else
438 W   T   T    * initialize record-identifying variables
439 W   T   T    store dfqtr to vqtr
440 W   T   T    store dfyear to vyear
441 W   T   T    store space(2) to       nprov
442 W   T   T    store space(4) to       ndist
443 W   T   T    store space(3) to nplace
444 W   T   T    store space(1) to       nala,nalb,nalc
445 W   T   T    store space(2) to nald
446 W   T   T    store dfconno to ncon
447 W   T   T    store space(2) to       vseq
448 W   T   T    store space(2) to vsubgrant
449 ^============LOOP
450 W   T    ELSE
451 <============EXIT && main loop
452 W   T    ENDIF
453 W    ENDIF && process=delete.  *** END OF DELETE RECORD PROCESS ***
454 W
455 W    *** GET REST OF SCREEN ***
456 W
457 W    IF .not. dfprice .and.('B'$(ald->indic).or.'C'$(ald->indic))
458 W   T    @ 13, 56  say 'Donated?' get vdonated pict 'Y'
459 W   T    read
460 W   T    IF lastkey()=27
461 <------------return
462 W   T    ENDIF
```

```
463 W     ENDIF
464 W     @ 15, 2  get  vqty pict '9999'
465 W     IF .not. dfprice
466 W   T     IF ald->vwamt
467 W   T   T     @ 17, 23  get  vaidamt pict '99999999'
468 W   T   T     read
469 W   T   T     IF lastkey()=27
470 <-----------------return
471 W   T   T     ENDIF
472 W   T     ENDIF
473 W   T     IF ald->vwtrnsp
474 W   T   T     @ 18, 23  get  vtransport pict '99999999'
475 W   T   T     read
476 W   T   T     IF lastkey()=27
477 <-----------------return
478 W   T   T     ENDIF
479 W   T     ENDIF
480 W   T     IF ald->vwdates
481 W   T   T     @ 20, 23  get  vbegin
482 W   T   T     read
483 W   T   T     IF lastkey()=27
484 <-----------------return
485 W   T   T     ENDIF
486 W   T   T     @ 21, 23  get  vend
487 W   T   T     read
488 W   T   T     IF lastkey()=27
489 <-----------------return
490 W   T   T     ENDIF
491 W   T     ENDIF
492 W   T     IF ald->vwverif
493 W   T   T     DO WHILE .T.
494 W   T   T   W     @ 22,  9  get  vverstat pict '99'
495 W   T   T   W     read
496 W   T   T   W     IF lastkey()=27
497 <-----------------------return
498 W   T   T   W     ENDIF
499 W   T   T   W     IF vverstat>=-1.and.vverstat<=2
500 W   T   T<===========EXIT
501 W   T   T   W     ELSE
502 W   T   T   W   F     ? chr(7)
503 W   T   T   W     ENDIF
504 W   T   T     ENDDO
505 W   T   T     @ 22, 23  get  vverdate
506 W   T   T     read
507 W   T   T     IF lastkey()=27
508 <-----------------return
509 W   T   T     ENDIF
510 W   T     ENDIF
511 W   T     IF ald->vwafghs
512 W   T   T     @ 15, 72  get  vmafsbene pict '999999'
513 W   T   T     read
514 W   T   T     IF lastkey()=27
515 <-----------------return
516 W   T   T     ENDIF
517 W   T   T     @ 16, 72  get  vfafsbene pict '999999'
518 W   T   T     read
519 W   T   T     IF lastkey()=27
520 <-----------------return
521 W   T   T     ENDIF
522 W   T     ENDIF
523 W   T     IF ald->vwnafghs
```

```
524 W   T   T    @ 17, 72  get  vnafsbene pict '999999'
525 W   T   T    read
526 W   T   T    IF lastkey()=27
527 <-----------------return
528 W   T   T    ENDIF
529 W   T    ENDIF
530 W   T    IF ald->vwunsal
531 W   T   T    @ 18, 70  get  vunsbene  pict '99999999'
532 W   T   T    read
533 W   T   T    IF lastkey()=27
534 <-----------------return
535 W   T   T    ENDIF
536 W   T    ENDIF
537 W   T    IF ald->vwindir
538 W   T   T    @ 19, 70  get  vindbene  pict '99999999'
539 W   T   T    read
540 W   T   T    IF lastkey()=27
541 <-----------------return
542 W   T   T    ENDIF
543 W   T    ENDIF
544 W   T    IF ald->vwverif
545 W   T   T    @ 21, 48  get  vintermed
546 W   T   T    read
547 W   T   T    IF lastkey()=27
548 <-----------------return
549 W   T   T    ENDIF
550 W   T   T    IF vpartyno=space(2)
551 W   T   T   T    DO CHPARTY
552 W   T   T   T    IF lastkey()=27
553 <-----------------------return
554 W   T   T   T    ENDIF
555 W   T   T    ELSE
556 W   T   T   F    dummy=vpartyno+' < '+trim(vparty)+'?'
557 W   T   T   F    answer='Y'
558 W   T   T   F    @ 22,41 say dummy get answer pict 'Y'
559 W   T   T   F    read
560 W   T   T   F    IF answer='Y'
561 W   T   T   F   T    @ 22,41 say vpartyno+' < '+trim(vparty)+space(4)
562 W   T   T   F    ELSE
563 W   T   T   F   F    DO CHPARTY
564 W   T   T   F   F    IF lastkey()=27
565 <---------------------------return
566 W   T   T   F   F    ENDIF
567 W   T   T   F    ENDIF
568 W   T   T    ENDIF
569 W   T    ENDIF
570 W    ENDIF   && .not.dfprice *** END OF GETS FOR REST OF SCREEN ****
571 W    answer='Y'
572 W    @ 24,24 say 'Is this screen correct (Y/N)?' get answer pict 'Y'
573 W    read
574 W    IF lastkey()=27
575 <--------return
576 W    ENDIF
577 W
578 W    *** REPLACE IF ENTER OR MODIFY ********************
579 W    IF answer='Y'   && screen correct: append blank or change record;;
579 W     ask if exit
580 W   T    select repdata
581 W   T    IF process='E'
582 W   T   T    append blank
583 W   T   T    replace update with date()
```

```
584 W   T    ENDIF
585 W   T    IF process='M'
586 W   T   T    IF dpi<>vdpi.or.qty<>vqty.or.aidamt<>vaidamt.or.transport<>;
586 W   T   T     vtransport.or.mafsbene<>vmafsbene.or.fafsbene<>vfafsbene.or.;
586 W   T   T     nafsbene<>vnafsbene.or.unsbene<>vunsbene.or.indbene<>vindben;
586 W   T   T    e .or.begin<>vbegin.or.end<>vend.or.verdate<>vverdate.or.;
586 W   T   T     verstat<>vverstat.or.intermed<>vintermed.or.partyno<>vpartyn;
586 W   T   T    o .or.donated<>vdonated
587 W   T   T   T    replace update with date()
588 W   T   T   T    vupdate=date()  && necessary only if same record edited;
588 W   T   T   T     next time
589 W   T   T    ENDIF
590 W   T    ENDIF
591 W   T    replace qtr with vqtr, year with vyear, provno with nprov,;
591 W   T     distno with ndist, placeno with nplace, ala with nala, alb with;
591 W   T      nalb, alc with nalc, ald with nald, conno with ncon, seq with;
591 W   T     vseq, subgrant with vsubgrant, dpi with vdpi, qty with vqty,;
591 W   T     aidamt with vaidamt, transport with vtransport
592 W   T    replace mafsbene with vmafsbene, fafsbene with vfafsbene, ;
592 W   T     nafsbene with vnafsbene, unsbene with vunsbene, indbene with;
592 W   T     vindbene, begin with vbegin, end with vend, verdate with;
592 W   T     vverdate, verstat with vverstat, intermed with vintermed,;
592 W   T    partyno with vpartyno,donated with vdonated, price with vprice
593 W   T    IF ncon<>space(2)
594 W   T   T    replace conabbrev with vconabbrev
595 W   T    ENDIF
596 W   T    @ 24,0 clear
597 W   T    answer='Y'
598 W   T    IF process='E'
599 W   T   T    @ 24,25 say 'Enter another record?' get answer pict 'Y'
600 W   T    ELSE
601 W   T   F    @ 24,25 say 'Modify another record?' get answer pict 'Y'
602 W   T    ENDIF
603 W   T    read
604 W   T    IF answer='N'
605 <============EXIT
606 W   T    ENDIF
607 W   T    firstpass=.T.
608 W    ENDIF
609 W
610 W    **  SET MATCHING VARIABLES FOR TEST NEXT TIME THROUGH SCREEN
611 W    @ 10,2 clear to 10,40
612 W    @  6,ccs-2 say space(1)
613 W    @  7,ccs-2 say space(1)
614 W    @  8,ccs-2 say space(1)
615 W    @  9,ccs-3 say space(2)
616 W    @ 12,ccs-3 say space(2)
617 W    @ 13,ccs-3 say space(2)
618 W    IF process='M' .and. firstpass
619 W   T    mqtr=vqtr
620 W   T    myear=vyear
621 W   T    mprov=nprov
622 W   T    mdist=ndist
623 W   T    mplace=nplace
624 W   T    mala=nala
625 W   T    malb=nalb
626 W   T    malc=nalc
627 W   T    mald=nald
628 W   T    mcon=ncon
629 W   T    msubgrant=vsubgrant
630 W   T    mseq=vseq
```

```
631 W    T     firstpass=.F.
632 W     ENDIF
633 ENDDO
634 IF packdb
635 T     select repdata
636 T     pack
637 ENDIF
638 close all
639 return
640
641
```

++ End of Program ENTRY ++ DOC found no errors.

VARIABLES: LOCAL CONCORDANCE FOR ENTRY

Variable	Statement number in listing (modified variables indicated by '=')										
AIDAMT:	298	586	=591								
ALA:	254	=591									
ALB:	254	=591									
ALC:	254	=591									
ALD:	254	=591									
answer:	=90	=91	94	96	99	102	=106	=163	=167	169	=407
	=408	410	=412	=414	417	=421	=423	=426	=428	=432	=434
	437	=557	=558	560	=571	=572	579	=597	=599	=601	604
BEGIN:	305	586	=592								
cala:	=69	185	208								
calb:	=69	186	209								
calc:	=69	187	210								
cald:	=69	188	211								
ccon:	=70	212									
ccs:	=80	81	129	130	131	132	133	134	135	137	162
	181	182	183	184	185	186	187	188	204	205	206
	207	208	209	210	211	212	612	613	614	615	616
	617										
cct:	=81										
cdist:	=67	183	206								
CONABBREV:	41	=594									
CONNO:	254	=591									
cplace:	=68	184	207								
cprov:	=66	182	205								
df*:	173										
dfconname:	70										
dfconno:	40	62	198	446							
dfprice:	136	160	=171	189	249	361	457	465			
dfqtr:	55	439									
dfsubgrant:	213										
dfyear:	56	440									
DISTNO:	254	=591									
DONATED:	296	586	=592								
DPI:	295	586	=591								
dummy:	=556	558									
END:	306	586	=592								
FAFSBENE:	301	586	=592								
firstpass:	=144	277	288	293	=607	618	=631				
INDBENE:	304	586	=592								
INDIC:	367	457									
INTERMED:	309	586	=592								
MAFSBENE:	300	586	=592								
mala:	=60	288	328	=624							
malb:	=60	288	328	=625							

VARIABLES: LOCAL CONCORDANCE FOR ENTRY (continued)

Variable	Statement number in listing (modified variables indicated by '=')										
malc:	=60	288	328	=626							
mald:	=61	288	328	=627							
mcon:	=62	288	328	=628							
mdist:	=58	288	328	=622							
MEAS:	364										
mplace:	=59	288	328	=623							
mprov:	=57	288	328	=621							
mqtr:	=55	288	328	=619							
mseq:	=63	288	328	=630							
msubgrant:	=64	288	328	=629							
myear:	=56	288	328	=620							
n:	411	416									
NAFSBENE:	302	586	=592								
nala:	=60	=132	252	254	288	292	=444	591	624		
nalb:	=60	=133	252	254	288	292	=444	591	625		
nalc:	=60	=134	252	254	288	292	=444	591	626		
nald:	=61	=135	252	254	288	292	=445	591	627		
ncon:	=62	=137	212	252	254	288	292	=446	591	593	628
ndist:	=58	=130	252	254	288	292	=442	591	622		
nplace:	=59	=131	252	254	288	292	=443	591	623		
nprov:	=57	=129	252	254	288	292	=441	591	621		
off:	172										
on:	174										
packdb:	=145	=420	634								
PARTY:	315										
PARTYNO:	311	314	586	=592							
PLACENO:	254	=591									
PRICE:	=592										
PRICEUNIT:	160	362									
process:	=97	=100	=103	251	276	288	406	581	585	598	618
PROVNO:	254	=591									
QTR:	254	=591									
QTY:	297	586	=591								
SEQ:	258	=591									
SUBGRANT:	254	=591									
times:	=331	333	=334								
TRANSPORT:	299	586	=591								
UNSBENE:	303	586	=592								
UPDATE:	310	=583	=587								
vaidamt:	=73	=279	=298	=371	=467	586	591				
vbegin:	=74	=280	=305	=377	=481	586	592				
vconabbrev:	=41	594									
vdonated:	=72	=278	=296	=368	=458	586	592				
vdpi:	=72	=191	=278	=295	586	591					
vend:	=74	=280	=306	=378	=486	586	592				
VERDATE:	307	586	=592								
VERSTAT:	308	586	=592								
vfafsbene:	=73	=279	=301	=386	=517	586	592				
vindbene:	=73	=279	=304	=395	=538	586	592				
vintermed:	=78	=284	=309	=398	=545	586	592				
vmafsbene:	=73	=279	=300	=385	=512	586	592				
vnafsbene:	=73	=279	=302	=389	=524	586	592				
vparty:	=77	=283	=315	556	561						
vpartyno:	=76	=282	=311	=401	550	556	561	586	592		
vprice:	=72	=139	=170	592							
vqtr:	=55	228	229	=232	234	235	=240	252	254	288	292
	=439	591	619								

VARIABLES: LOCAL CONCORDANCE FOR ENTRY (continued)

Variable	Statement number in listing (modified variables indicated by '=')										
vqty:	=73	=279	=297	=360	=464	586	591				
vseq:	=63	=258	=260	262	=264	=265	=270	288	292	=447	591
	630										
vsubgrant:	=64	=215	252	254	288	292	=448	591	629		
vtransport:	=73	=279	=299	=374	=474	586	591				
vunsbene:	=73	=279	=303	=392	=531	586	592				
vupdate:	=79	=285	=310	345	=588						
vverdate:	=74	=280	=307	=382	=505	586	592				
vverstat:	=75	=281	=308	=381	=494	499	586	592			
VWAFGHS:	384	511									
VWAMT:	370	466									
VWDATES:	376	480									
VWDPI:	190										
VWINDIR:	394	537									
VWNAFGHS:	388	523									
VWTRNSP:	373	473									
VWUNSAL:	391	530									
VWVERIF:	380	397	400	492	544						
vyear:	=56	228	229	230	231	=236	=241	252	254	288	292
	=440	591	620								
w:	411	416									
YEAR:	254	=591									

```
 1 * default.prg - set defaults for system
 2 * ver 1.0 Ron Barney 11 November 1990
 3
 4 ** global default variables stored in pantry.mem:
 5 ** dfconno, dfconname, dfqtr, dfyear, dfprice, dfsubgrant
 6 ** restore from pantry.mem && to get these in program test.
 7
 8 use contract index conno
 9 clear
10 @  1,20 say 'USAID/Representative for Afghanistan/DC&A'
11 @  2,29 say 'PERFORMANCE DATA SYSTEM'
12 @  3,1 to 3,78
13 @  6,15 say 'Defaults:'
14 @  8,15 say '1 - Current quarter'
15 @  8,48 say dfqtr
16 @ 10,15 say '2 - Current year'
17 @ 10,48 say dfyear
18 @ 12,15 say '3 - Contractor number'
19 IF dfconno<>space(2)
20 T    @ 12,48 say dfconno+' ('+trim(dfconabbrev)+')'
21 ENDIF
22 @ 14,15 say '4 - Display prices, not units'
23 @ 14,48 say IIF(dfprice,'Y','N')
24 @ 16,15 say '5 - Display subgrant number'
25 @ 16,48 say IIF(dfsubgrant,'Y','N')
26 DO WHILE .T.
27 W    answer='0'
28 W    @ 19,15 say 'Selection:' get answer pict '!'
29 W    read
30 W    DO CASE
31 W    CASE answer='0'
32 <========EXIT    && *****
33 W    CASE answer='1' && current quarter
34 W   2    DO WHILE .T.
35 W   2   W    @ 8,48 get dfqtr
36 W   2   W    read
37 W   2   W    IF at(dfqtr,'1234 ')<>0
38 W   2   W   T    @  8,48 say dfqtr
39 W   2<===========EXIT
40 W   2   W    ELSE
41 W   2   W   F    ? chr(7)
42 W   2   W    ENDIF
43 W   2    ENDDO
44 W    CASE answer='2' && current year
45 W   3    DO WHILE .T.
46 W   3   W    @ 10,48 get dfyear
47 W   3   W    read
48 W   3   W    IF dfyear=space(2).or.val(dfyear)>0
49 W   3   W   T    @ 10,48 say dfyear
50 W   3<===========EXIT
51 W   3   W    ELSE
52 W   3   W   F    ? chr(7)
53 W   3   W    ENDIF
54 W   3    ENDDO
55 W    CASE answer='3' && contractor number
56 W   4    DO WHILE .T.
57 W   4   W    @ 12,48 get dfconno
58 W   4   W    read
59 W   4   W    IF len(trim(dfconno))=1
60 W   4   W   T    dfconno=subst(str(val(dfconno)+100,3),2)
61 W   4   W    ENDIF
```

```
62 W    4    W     seek dfconno
63 W    4    W     IF found()
64 W    4    W    T     dfconabbrev=conabbrev
65 W    4    W    T     dfconname=conname
66 W    4    W    T     @ 12,48 say dfconno+' ('+trim(dfconabbrev)+')'+space(13)
67 W    4<============EXIT && do while to get dfconno
68 W    4    W     ELSE
69 W    4    W    F     IF dfconno=space(2)
70 W    4    W    F    T     dfconabbrev=space(13)
71 W    4    W    F    T     @ 12,48 say space(18)
72 W    4<================EXIT    && do while to get dfconno
73 W    4    W    F     ELSE
74 W    4    W    F    F     ? chr(7)
75 W    4    W    F     ENDIF
76 W    4    W     ENDIF
77 W    4     ENDDO
78 W     CASE answer='4' && display prices not units
79 W    5     @ 14,48 get dfprice pict 'Y'
80 W    5     read
81 W    5     @ 14,48 say IIF(dfprice,'Y','N')
82 W     CASE answer='5' && display subgrant number
83 W    6     @ 16,48 get dfsubgrant pict 'Y'
84 W    6     read
85 W    6     @ 16,48 say IIF(dfsubgrant,'Y','N')
86 W     ENDCASE
87 ENDDO
88 set safety off
89 save all like df* to pantry.mem
90 set safety on
91
92 close all
93 return
94
95
96
97
98
99
```

++ End of Program DEFAULT ++ DOC found no errors.

VARIABLES: LOCAL CONCORDANCE FOR DEFAULT

Variable	Statement number in listing (modified variables indicated by '=')							
answer:	=27	=28	31	33	44	55	78	82
CONABBREV:	64							
CONNAME:	65							
df*:	89							
dfconabbre:	20	=64	66	=70				
dfconname:	=65							
dfconno:	19	20	=57	59	=60	62	66	69
dfprice:	23	=79	81					
dfqtr:	15	=35	37	38				
dfsubgrant:	25	=83	85					
dfyear:	17	=46	48	49				
off:	88							
on:	90							

```
1 * rptobj.prg - full report of indicators by objective
2 * ver 1.0 Ron Barney 18 November 1990
3 * Print by objective indicator number/by province in alpha order/
4 * by district in alpha order/by indicator grouping at the third level,
5 * in code order/by indicator, in code order.  Provide totals by
6 * district, province and for report as a whole.
7
8 select 1
9 use indic index indno && key indno
10 select 2
11 use province index province     && keys province
12 set filter to provno<>'00'                && don't print unspecified;
12  locations
13 go top
14 select 3
15 use district index pdistrict    && keys provno+district
16 select 4
17 use ala index ala       && key ala
18 select 5
19 use alb index alb       && key ala+alb
20 set relation to ala into ala
21 select 6
22 use alc index alc       && key ala+alb+alc
23 set relation to ala+alb into alb
24 select 7
25 use ald index ald       && key ala+alb+alc+ald
26
27 @ 4,0 clear
28 @  6,20 say 'Semiannual Indicators by Objective:'
29 qtrset=space(1)
30 yearset=space(2)
31 @  9,20 say 'Which quarter? ' get qtrset pict '9'
32 @ 11,20 say 'Which year?   ' get yearset pict '99'
33 read
34 answer='N'
35 @ 13,20 say 'If you have changed data since the last'
36 @ 14,20 say 'printing, you must reindex repdata.dbf so'
37 @ 15,20 say 'that your additions will be printed.'
38 @ 16,20 say 'Do you want to reindex?' get answer pict 'Y'
39 read
40
41 select 8
42 use repdata index rpt1  && key provno+distno+ala+alb+alc+ald+conabbrev
43 IF answer='Y'
44 T    @ 19,20 say 'Reindexing...'
45 T    reindex
46 ENDIF
47 set relation to ala+alb+alc+ald into ald
48 @ 19,20 say 'Printing...  '
49
50 ** initialize totals for district (dt), province(pt) and report (rt)
51 store 0 to dtaidamt,dttranspor,dtmafsbene,dtfafsbene,dtnafsbene,dtunsbene,;
51  dtindbene
52 store 0 to ptaidamt,pttranspor,ptmafsbene,ptfafsbene,ptnafsbene,ptunsbene,;
52  ptindbene
53 store 0 to rtaidamt,rttranspor,rtmafsbene,rtfafsbene,rtnafsbene,rtunsbene,;
53  rtindbene
54 rr=1    && row counter
55 pp=1    && page counter
56
57 set margin to 0
```

```
58 set device to print
59
60 ** PROCESS EACH INDICATOR:
61 pass1=.T.
62 select indic
63 DO WHILE .not. eof()
64 W
65 W     IF pass1
66 W    T    DO RPTOBJHD      && print header
67 W    T    newind=.F.       && print name of new indicator
68 W     ELSE
69 W    F    newind=.T.       && print name of new indicator
70 W     ENDIF
71 W     select repdata
72 W     DO CASE
73 W     CASE indic->indno='A'    && market prices
74 W    1    set filter to qtr=qtrset.and.year=yearset.and.price
75 W     CASE indic->indno='B'    && donated
76 W    2    set filter to;
76 W    2     qtr=qtrset.and.year=yearset.and.donated.and..not.price
77 W     CASE indic->indno='C'
78 W    3    set filter to qtr=qtrset.and.year=yearset.and..not.donated;
78 W    3     .and..not.price
79 W     CASE indic->indno='J'.or.indic->indno='K'
80 W    4    set filter to qtr=qtrset.and.year=yearset.and.dpi
81 W     OTHERWISE
82 W    0    set filter to qtr=qtrset.and.year=yearset
83 W     ENDCASE
84 W     go top
85 W
86 W     ** PROCESS EACH PROVINCE:
87 W     select province
88 W     provtot=.F.              && process province totals after print
89 W     go top
90 W     DO WHILE .not. eof()
91 W    W
92 W    W    ** skip to next province if no records to print for this province
93 W    W    select repdata
94 W    W    seek province->provno
95 W    W    IF found()
96 W    W   T    newprov=.T.      && print name of new province
97 W    W   T    ** PROCESS EACH DISTRICT IN PROVINCE:
98 W    W   T    select district && key provno+district name
99 W    W   T    disttot=.F.                        && process district totals;
99 W    W   T     after print
100 W   W   T    seek province->provno   && position at first district of;
100 W   W   T     province
101 W   W   T    DO WHILE district->provno=province->provno
102 W   W   T   W
103 W   W   T   W    ** skip to next district if no records to print for this
103 W   W   T   W    *  district
104 W   W   T   W    select repdata
105 W   W   T   W    seek district->provno+district->distno
106 W   W   T   W    IF found()
107 W   W   T   W   T    newdist=.T.      && print name of new district
108 W   W   T   W   T    ** PROCESS EACH 3rd-LEVEL CODE COMBINATION:
109 W   W   T   W   T    select alc
110 W   W   T   W   T    go top
111 W   W   T   W   T    DO WHILE .not. eof()
112 W   W   T   W   T   W
113 W   W   T   W   T   W    newalc=.T.       && print name of new 3rd-level
```

```
113 W   W   T   W   T   W    code
114 W   W   T   W   T   W
115 W   W   T   W   T   W    ** PRINT/SUMMARIZE EACH REPDATA RECORD (IN CODE
115 W   W   T   W   T   W    *  ORDER)
116 W   W   T   W   T   W    ** FOR CURRENT INDICATOR, PLACE AND 3RD-LEVEL
116 W   W   T   W   T   W    *  CODE:
117 W   W   T   W   T   W    setdata=province->provno+district->distno+ala+;
117 W   W   T   W   T   W    alb+alc
118 W   W   T   W   T   W    select repdata
119 W   W   T   W   T   W    seek setdata
120 W   W   T   W   T   W    IF found()
121 W   W   T   W   T   W  T    DO WHILE provno+distno+ala+alb+alc=setdata
122 W   W   T   W   T   W  T  W    IF indic->indno$(ald->indic)
123 W   W   T   W   T   W  T  W  T    store .T. to provtot, disttot   &&;
123 W   W   T   W   T   W  T  W  T     set up totals
124 W   W   T   W   T   W  T  W  T    IF rr>55 .or. newind
125 W   W   T   W   T   W  T  W  T  T    eject
126 W   W   T   W   T   W  T  W  T  T    DO RPTOBJHD      && page header
127 W   W   T   W   T   W  T  W  T  T    newind=.F.
128 W   W   T   W   T   W  T  W  T    ENDIF
129 W   W   T   W   T   W  T  W  T    IF newprov       && print new;
129 W   W   T   W   T   W  T  W  T     province name
130 W   W   T   W   T   W  T  W  T  T    IF provno<>'99'
131 W   W   T   W   T   W  T  W  T  T  T    @ rr,1 say;
131 W   W   T   W   T   W  T  W  T  T  T     upper(trim(province->provinc;
131 W   W   T   W   T   W  T  W  T  T  T    e))+ ' PROVINCE'
132 W   W   T   W   T   W  T  W  T  T    ELSE
133 W   W   T   W   T   W  T  W  T  T  F    @ rr,1 say 'PAKISTAN'
134 W   W   T   W   T   W  T  W  T  T    ENDIF
135 W   W   T   W   T   W  T  W  T  T    rr=rr+1
136 W   W   T   W   T   W  T  W  T  T    newprov=.F.
137 W   W   T   W   T   W  T  W  T    ENDIF
138 W   W   T   W   T   W  T  W  T    IF newdist       && print new;
138 W   W   T   W   T   W  T  W  T     district name
139 W   W   T   W   T   W  T  W  T  T    @ rr,3 say;
139 W   W   T   W   T   W  T  W  T  T     trim(district->district)+ ;
139 W   W   T   W   T   W  T  W  T  T    ' District'
140 W   W   T   W   T   W  T  W  T  T    rr=rr+1
141 W   W   T   W   T   W  T  W  T  T    newdist=.F.
142 W   W   T   W   T   W  T  W  T    ENDIF
143 W   W   T   W   T   W  T  W  T    IF newalc        && print new 3rd-;
143 W   W   T   W   T   W  T  W  T    level code name
144 W   W   T   W   T   W  T  W  T  T    @ rr,5 say trim(ala->alaname)+;
144 W   W   T   W   T   W  T  W  T  T    '/'+ trim(alb->albname)+'/'+;
144 W   W   T   W   T   W  T  W  T  T    alc->alcname
145 W   W   T   W   T   W  T  W  T  T    rr=rr+1
146 W   W   T   W   T   W  T  W  T  T    newalc=.F.
147 W   W   T   W   T   W  T  W  T    ENDIF
148 W   W   T   W   T   W  T  W  T    ** print a line
149 W   W   T   W   T   W  T  W  T    @ rr,7 say ald->aldname
150 W   W   T   W   T   W  T  W  T    @ rr,39 say conabbrev
151 W   W   T   W   T   W  T  W  T    @ rr,54 say qty pict '9999'
152 W   W   T   W   T   W  T  W  T    IF indic->indno<>'A'
153 W   W   T   W   T   W  T  W  T  T    @ rr,60 say ald->meas
154 W   W   T   W   T   W  T  W  T    ELSE
155 W   W   T   W   T   W  T  W  T  F    @ rr,60 say ald->priceunit
156 W   W   T   W   T   W  T  W  T    ENDIF
157 W   W   T   W   T   W  T  W  T    IF aidamt>0
158 W   W   T   W   T   W  T  W  T  T    @ rr,76 say aidamt pict ;
158 W   W   T   W   T   W  T  W  T  T    '999999999'
159 W   W   T   W   T   W  T  W  T    ENDIF
```

```
160 W   W   T   W   T   W   T   W   T    IF transport>0
161 W   W   T   W   T   W   T   W   T   T    @ rr,88 say transport pict ;
161 W   W   T   W   T   W   T   W   T   T    '9999999'
162 W   W   T   W   T   W   T   W   T    ENDIF
163 W   W   T   W   T   W   T   W   T    IF mafsbene>0
164 W   W   T   W   T   W   T   W   T   T    @ rr,98 say mafsbene pict ;
164 W   W   T   W   T   W   T   W   T   T    '999999'
165 !   W   T   W   T   W   T   W   T    ENDIF
166 W   W   T   W   T   W   T   W   T    IF fafsbene>0
167 W   W   T   W   T   W   T   W   T   T    @ rr,106 say fafsbene pict ;
167 W   W   T   W   T   W   T   W   T   T    '999999'
168 W   W   T   W   T   W   T   W   T    ENDIF
169 W   W   T   W   T   W   T   W   T    IF nafsbene>0
170 W   W   T   W   T   W   T   W   T   T    @ rr,114 say nafsbene pict ;
170 W   W   T   W   T   W   T   W   T   T    '999999'
171 W   W   T   W   T   W   T   W   T    ENDIF
172 W   W   T   W   T   W   T   W   T    IF mafsbene+fafsbene+nafsbene>0
173 W   W   T   W   T   W   T   W   T   T    @ rr,124 say mafsbene+fafsbene+;
173 W   W   T   W   T   W   T   W   T   T    nafsbene pict '999999'
174 W   W   T   W   T   W   T   W   T    ENDIF
175 W   W   T   W   T   W   T   W   T    IF mafsbene+fafsbene+nafsbene>0
176 W   W   T   W   T   W   T   W   T   T    @ rr,133 say round((mafsbene+;
176 W   W   T   W   T   W   T   W   T   T    fafsbene)/ (mafsbene+fafsbene+;
176 W   W   T   W   T   W   T   W   T   T    nafsbene)*100,0) pict '999'+'%'
177 W   W   T   W   T   W   T   W   T    ENDIF
178 W   W   T   W   T   W   T   W   T    IF unsbene>0
179 W   W   T   W   T   W   T   W   T   T    @ rr,141 say unsbene pict ;
179 W   W   T   W   T   W   T   W   T   T    '9999999'
180 W   W   T   W   T   W   T   W   T    ENDIF
181 W   W   T   W   T   W   T   W   T    IF indbene>0
182 W   W   T   W   T   W   T   W   T   T    @ rr,152 say indbene pict ;
182 W   W   T   W   T   W   T   W   T   T    '9999999'
183 W   W   T   W   T   W   T   W   T    ENDIF
184 W   W   T   W   T   W   T   W   T    IF begin<>ctod('  /  /  ')
185 W   W   T   W   T   W   T   W   T   T    @ rr,164 say begin
186 W   W   T   W   T   W   T   W   T    ENDIF
187 W   W   T   W   T   W   T   W   T    IF end<>ctod('  /  /  ')
188 W   W   T   W   T   W   T   W   T   T    @ rr,174 say end
189 W   W   T   W   T   W   T   W   T    ENDIF
190 W   W   T   W   T   W   T   W   T    IF (aidamt+transport+mafsbene+;
190 W   W   T   W   T   W   T   W   T    fafsbene+nafsbene+ unsbene+indbene)>0
191 W   W   T   W   T   W   T   W   T   T    @ rr,185 say verstat
192 W   W   T   W   T   W   T   W   T    ENDIF
193 W   W   T   W   T   W   T   W   T    IF verdate<>ctod('  /  /  ')
194 W   W   T   W   T   W   T   W   T   T    @ rr,189 say verdate
195 W   W   T   W   T   W   T   W   T    ENDIF
196 W   W   T   W   T   W   T   W   T    rr=rr+1
197 W   W   T   W   T   W   T   W   T    ** increment district totals
198 W   W   T   W   T   W   T   W   T    dtaidamt=dtaidamt+aidamt
199 W   W   T   W   T   W   T   W   T    dttranspor=dttranspor+transport
200 W   W   T   W   T   W   T   W   T    dtmafsbene=dtmafsbene+mafsbene
201 W   W   T   W   T   W   T   W   T    dtfafsbene=dtfafsbene+fafsbene
202 W   W   T   W   T   W   T   W   T    dtnafsbene=dtnafsbene+nafsbene
203 W   W   T   W   T   W   T   W   T    dtunsbene=dtunsbene+unsbene
204 W   W   T   W   T   W   T   W   T    dtindbene=dtindbene+indbene
205 W   W   T   W   T   W   T   W    ENDIF   && indic->indno$(ald->indic)
206 W   W   T   W   T   W   T   W    skip
207 W   W   T   W   T   W   T    ENDDO   && provno+distno+ala+alb+alc=setdata
208 W   W   T   W   T   W    ENDIF   && alc found
209 W   W   T   W   T   W    select alc
210 W   W   T   W   T   W    skip
```

```
211 W   W   T   W   T    ENDDO   && .not. eof() in alc.dbf
212 W   W   T   W    ENDIF
213 W   W   T   W    IF disttot
214 W   W   T   W   T    ** print district totals:
215 W   W   T   W   T    @ rr,76 say repl('-',83)
216 W   W   T   W   T    rr=rr+1
217 W   W   T   W   T    @ rr,5 say 'District Totals'
218 W   W   T   W   T    @ rr,76 say dtaidamt pict '999999999'
219 W   W   T   W   T    @ rr,88 say dttranspor pict '9999999'
220 W   W   T   W   T    @ rr,98 say dtmafsbene pict '999999'
221 W   W   T   W   T    @ rr,106 say dtfafsbene pict '999999'
222 W   W   T   W   T    @ rr,114 say dtnafsbene pict '999999'
223 W   W   T   W   T    @ rr,124 say dtmafsbene+dtfafsbene+dtnafsbene pict ;
223 W   W   T   W   T    '999999'
224 W   W   T   W   T    IF dtmafsbene+dtfafsbene+dtnafsbene>0
225 W   W   T   W   T   T    @ rr,133 say round((dtmafsbene+dtfafsbene)/;
225 W   W   T   W   T   T     (dtmafsbene+dtfafsbene+dtnafsbene)*100,0) pict ;
225 W   W   T   W   T   T    '999'+'%'
226 W   W   T   W   T    ENDIF
227 W   W   T   W   T    @ rr,141 say dtunsbene pict '9999999'
228 W   W   T   W   T    @ rr,152 say dtindbene pict '9999999'
229 W   W   T   W   T    rr=rr+1
230 W   W   T   W   T    ** add district totals to province totals:
231 W   W   T   W   T    ptaidamt=ptaidamt+dtaidamt
232 W   W   T   W   T    pttranspor=pttranspor+dttranspor
233 W   W   T   W   T    ptmafsbene=ptmafsbene+dtmafsbene
234 W   W   T   W   T    ptfafsbene=ptfafsbene+dtfafsbene
235 W   W   T   W   T    ptnafsbene=ptnafsbene+dtnafsbene
236 W   W   T   W   T    ptunsbene=ptunsbene+dtunsbene
237 W   W   T   W   T    ptindbene=ptindbene+dtindbene
238 W   W   T   W   T    ** reinitialize district accumulators:
239 W   W   T   W   T    store 0 to dtaidamt,dttranspor,dtmafsbene,dtfafsbene,;
239 W   W   T   W   T     dtnafsbene,dtunsbene,dtindbene
240 W   W   T   W   T    disttot=.F.
241 W   W   T   W    ENDIF
242 W   W   T   W    select district
243 W   W   T   W    skip
244 W   W   T    ENDDO   && district->provno=province->provno in district.dbf
245 W   W   T    IF provtot
246 W   W   T   T    ** print province totals:
247 W   W   T   T    @ rr,76 say repl('-',83)
248 W   W   T   T    rr=rr+1
249 W   W   T   T    IF province->provno<>'99'
250 W   W   T   T   T    @ rr,3 say 'PROVINCE TOTALS'
251 W   W   T   T    ELSE
252 W   W   T   T   F    @ rr,3 say 'PAKISTAN TOTALS'
253 W   W   T   T    ENDIF
254 W   W   T   T    @ rr,76 say ptaidamt pict '999999999'
255 W   W   T   T    @ rr,88 say pttranspor pict '9999999'
256 W   W   T   T    @ rr,98 say ptmafsbene pict '999999'
257 W   W   T   T    @ rr,106 say ptfafsbene pict '999999'
258 W   W   T   T    @ rr,114 say ptnafsbene pict '999999'
259 W   W   T   T    @ rr,124 say ptmafsbene+ptfafsbene+ptnafsbene pict ;
259 W   W   T   T    '999999'
260 W   W   T   T    IF ptmafsbene+ptfafsbene+ptnafsbene>0
261 W   W   T   T   T    @ rr,133 say round((ptmafsbene+ptfafsbene)/;
261 W   W   T   T   T     (ptmafsbene+ptfafsbene+ptnafsbene)*100,0) pict ;
261 W   W   T   T   T    '999'+'%'
262 W   W   T   T    ENDIF
263 W   W   T   T    @ rr,141 say ptunsbene pict '9999999'
264 W   W   T   T    @ rr,152 say ptindbene pict '9999999'
```

```
265 W    W    T    T     rr=rr+2
266 W    W    T    T     ** add province totals to report totals:
267 W    W    T    T     rtaidamt=rtaidamt+ptaidamt
268 W    W    T    T     rttranspor=rttranspor+pttranspor
269 W    W    T    T     rtmafsbene=rtmafsbene+ptmafsbene
270 W    W    T    T     rtfafsbene=rtfafsbene+ptfafsbene
271 W    W    T    T     rtnafsbene=rtnafsbene+ptnafsbene
272 W    W    T    T     rtunsbene=rtunsbene+ptunsbene
273 W    W    T    T     rtindbene=rtindbene+ptindbene
274 W    W    T    T     ** reinitialize province accumulators:
275 W    W    T    T     store 0 to ptaidamt,pttranspor,ptmafsbene,ptfafsbene,;
275 W    W    T    T     ptnafsbene, ptunsbene,ptindbene
276 W    W    T    T     provtot=.F.
277 W    W    T     ENDIF    && provtot
278 W    W     ENDIF    && province->provno .not. found()
279 W    W     select province
280 W    W     skip
281 W     ENDDO    && .not. eof() in province.dbf
282 W     select indic
283 W     skip
284 ENDDO    && .not. eof() in indic.dbf
285 ** print report totals
286 @ rr,1 say 'REPORT TOTALS'
287 @ rr,76 say repl('-',83)
288 rr=rr+1
289 @ rr,76 say rtaidamt pict '999999999'
290 @ rr,88 say rttranspor pict '9999999'
291 @ rr,98 say rtmafsbene pict '999999'
292 @ rr,106 say rtfafsbene pict '999999'
293 @ rr,114 say rtnafsbene pict '999999'
294 @ rr,124 say rtmafsbene+rtfafsbene+rtnafsbene pict '999999'
295 IF rtmafsbene+rtfafsbene+rtnafsbene>0
296 T     @ rr,133 say round((rtmafsbene+rtfafsbene)/ (rtmafsbene+rtfafsbene+;
296 T     rtnafsbene)*100,0) pict '999'+'%'
297 ENDIF
298 @ rr,141 say rtunsbene pict '9999999'
299 @ rr,152 say rtindbene pict '9999999'
300 rr=rr+1
301 @ rr,76 say repl('=',83)
302 eject
303 eject
304 set device to screen
305 close all
306 return
307
```

++ End of Program RPTOBJ ++ DOC found no errors.

VARIABLES: LOCAL CONCORDANCE FOR RPTOBJ

Variable	Statement number in listing (modified variables indicated by '=')				
AIDAMT:	157	158	190	198	
ALA:	20	23	47	117	121
ALANAME:	144				
ALB:	23	47	117	121	
ALBNAME:	144				
ALC:	47	117	121		
ALCNAME:	144				
ALD:	47				
ALDNAME:	149				
answer:	=34	=38	43		

VARIABLES: LOCAL CONCORDANCE FOR RPTOBJ (continued)

Variable	Statement number in listing (modified variables indicated by '='										
BEGIN:	184	185									
CONABBREV:	150										
DISTNO:	105	117	121								
DISTRICT:	139										
disttot:	=99	=123	213	=240							
DONATED:	76	78									
DPI:	80										
dtaidamt:	=51	=198	218	231	=239						
dtfafsbene:	=51	=201	221	223	224	225	234	=239			
dtindbene:	=51	=204	228	237	=239						
dtmafsbene:	=51	=200	220	223	224	225	233	=239			
dtnafsbene:	=51	=202	222	223	224	225	235	=239			
dttranspor:	=51	=199	219	232	=239						
dtunsbene:	=51	=203	227	236	=239						
END:	187	188									
FAFSBENE:	166	167	172	173	175	176	190	201			
INDBENE:	181	182	190	204							
INDIC:	122										
INDNO:	73	75	77	79	122	152					
MAFSBENE:	163	164	172	173	175	176	190	200			
MEAS:	153										
NAFSBENE:	169	170	172	173	175	176	190	202			
newalc:	=113	143	=146								
newdist:	=107	138	=141								
newind:	=67	=69	124	=127							
newprov:	=96	129	=136								
pass1:	=61	65									
pp:	=55										
PRICE:	74	76	78								
PRICEUNIT:	155										
print:	58										
PROVINCE:	131										
PROVNO:	12	94	100	101	105	117	121	130	249		
provtot:	=88	=123	245	=276							
ptaidamt:	=52	=231	254	267	=275						
ptfafsbene:	=52	=234	257	259	260	261	270	=275			
ptindbene:	=52	=237	264	273	=275						
ptmafsbene:	=52	=233	256	259	260	261	269	=275			
ptnafsbene:	=52	=235	258	259	260	261	271	=275			
pttranspor:	=52	=232	255	268	=275						
ptunsbene:	=52	=236	263	272	=275						
QTR:	74	76	78	80	82						
qtrset:	=29	=31	74	76	78	80	82				
QTY:	151										
rr:	=54	124	131	133	=135	139	=140	144	=145	149	150
	151	153	155	158	161	164	167	170	173	176	179
	182	185	188	191	194	=196	215	=216	217	218	219
	220	221	222	223	225	227	228	=229	247	=248	250
	252	254	255	256	257	258	259	261	263	264	=265
	286	287	=288	289	290	291	292	293	294	296	298
	299	=300	301								
rtaidamt:	=53	=267	289								
rtfafsbene:	=53	=270	292	294	295	296					
rtindbene:	=53	=273	299								
rtmafsbene:	=53	=269	291	294	295	296					
rtnafsbene:	=53	=271	293	294	295	296					
rttranspor:	=53	=268	290								

VARIABLES: LOCAL CONCORDANCE FOR RPTOBJ (continued)

Variable	Statement number in listing (modified variables indicated by '='						
rtunsbene:	=53	=272	298				
screen:	304						
setdata:	=117	119	121				
TRANSPORT:	160	161	190	199			
UNSBENE:	178	179	190	203			
VERDATE:	193	194					
VERSTAT:	191						
YEAR:	74	76	78	80	82		
yearset:	=30	=32	74	76	78	80	82

```
1 * rptloc.prg - full report of indicators by location
2 * ver 1.0 Ron Barney 20 November 1990
3 * Print by province in alpha order/by district in alpha order/
4 * by objective indicator number/by indicator grouping at the third level,
5 * in code order/by indicator, in code order.  Provide totals by
6 * district, province and for report as a whole.
7
8 select 1
9 use province index province      && keys province
10 set filter to provno<>'00'                && don't print unspecified;
10  locations
11 go top
12 select 2
13 use district index pdistrict    && keys provno+district
14
15 select 3
16 use indic index indno && key indno
17
18 select 4
19 use ala index ala        && key ala
20 select 5
21 use alb index alb        && key ala+alb
22 set relation to ala into ala
23 select 6
24 use alc index alc        && key ala+alb+alc
25 set relation to ala+alb into alb
26 select 7
27 use ald index ald        && key ala+alb+alc+ald
28
29 @ 4,0 clear
30 @  6,20 say 'Semiannual Indicators by Location:'
31 qtrset=space(1)
32 yearset=space(2)
33 @  9,20 say 'Which quarter? ' get qtrset pict '9'
34 @ 11,20 say 'Which year?    ' get yearset pict '99'
35 read
36 answer='N'
37 @ 13,20 say 'If you have changed data since the last'
38 @ 14,20 say 'printing, you must reindex repdata.dbf so'
39 @ 15,20 say 'that your additions will be printed.'
40 @ 16,20 say 'Do you want to reindex?' get answer pict 'Y'
41 read
42
43 select 8
44 use repdata index rpt1  && key provno+distno+ala+alb+alc+ald+conabbrev
45 IF answer='Y'
46 T    @ 19,20 say 'Reindexing...'
47 T    reindex
48 ENDIF
49 set relation to ala+alb+alc+ald into ald
50 @ 19,20 say 'Printing...  '
51
52 ** initialize totals for indicator (it), district (dt) and province (pt)
53 store 0 to itaidamt,ittranspor,itmafsbene,itfafsbene,itnafsbene,itunsbene,;
53  itindbene,itstata,itstatb,itstatc,itstatd
54 store 0 to dtaidamt,dttranspor,dtmafsbene,dtfafsbene,dtnafsbene,dtunsbene,;
54  dtindbene,dtstata,dtstatb,dtstatc,dtstatd
55 store 0 to ptaidamt,pttranspor,ptmafsbene,ptfafsbene,ptnafsbene,ptunsbene,;
55  ptindbene,ptstata,ptstatb,ptstatc,ptstatd
56 store .F. to provtot,disttot,indictot
57
```

```
 58 rr=1     && row counter
 59 pp=1     && page counter
 60
 61 set margin to 0
 62 set device to print
 63
 64 ***************************************************************
 65 ** PROCESS EACH PROVINCE FOR WHICH THERE ARE REPDATA RECORDS:
 66 passi=.T.        && flag for first printing of header
 67 select province
 68 DO WHILE .not. eof()
 69 W     select repdata
 70 W     seek province->provno
 71 W     IF .not. found()
 72 W    T     select province
 73 W    T     skip
 74 ^========loop
 75 W     ENDIF
 76 W     newprov=.T.
 77 W
 78 W     ** PROCESS EACH DISTRICT FOR WHICH THERE ARE REPDATA RECORDS:
 79 W     select district
 80 W     seek province->provno
 81 W     DO WHILE provno=province->provno .and..not. eof()
 82 W    W     select repdata
 83 W    W     seek province->provno+district->distno
 84 W    W     IF .not.found()
 85 W    W    T     select district
 86 W    W    T     skip
 87 W    ^========loop
 88 W    W     ENDIF
 89 W    W     newdist=.T.
 90 W    W     IF passi
 91 W    W    T     DO RPTLOCHD
 92 W    W     ENDIF
 93 W    W
 94 W    W     ** PROCESS EACH INDICATOR FOR WHICH THERE ARE REPDATA RECORDS:
 95 W    W     select indic
 96 W    W     go top
 97 W    W     DO WHILE .not. eof()
 98 W    W    W     select repdata
 99 W    W    W     DO CASE
100 W    W    W     CASE indic->indno='A'    && market prices
101 W    W    W    1     set filter to qtr=qtrset.and.year=yearset.and.price
102 W    W    W     CASE indic->indno='B'    && donated
103 W    W    W    2     set filter to;
103 W    W    W    2      qtr=qtrset.and.year=yearset.and.donated.and..not.price
104 W    W    W     CASE indic->indno='C'
105 W    W    W    3     set filter to;
105 W    W    W    3      qtr=qtrset.and.year=yearset.and..not.donated;
105 W    W    W    3      .and..not.price
106 W    W    W     CASE indic->indno='J'.or.indic->indno='K'
107 W    W    W    4     set filter to qtr=qtrset.and.year=yearset.and.dpi
108 W    W    W     OTHERWISE
109 W    W    W    0     set filter to qtr=qtrset.and.year=yearset
110 W    W    W     ENDCASE
111 W    W    W     go top
112 W    W    W     seek province->provno+district->distno
113 W    W    W     IF .not. found()
114 W    W    W    T     select indic
115 W    W    W    T     skip
```

```
116 W    W    ^========loop
117 W    W    W     ENDIF
118 W    W    W     newindic=.T.
119 W    W    W
120 W    W    W     ** PROCESS EACH 3rd-LEVEL CODE COMBINATION:
121 W    W    W     select alc
122 W    W    W     go top
123 W    W    W     DO WHILE .not. eof()
124 W    W    W    W     setdata=province->provno+district->distno+ala+alb+alc
125 W    W    W    W     select repdata
126 W    W    W    W     seek setdata
127 W    W    W    W     IF .not. found()
128 W    W    W    W    T     select alc
129 W    W    W    W    T     skip
130 W    W    W    ^========loop
131 W    W    W    W     ENDIF
132 W    W    W    W     newalc=.T.
133 W    W    W    W
134 W    W    W    W     ** PRINT/SUMMARIZE EACH REPDATA RECORD (IN CODE ORDER)
135 W    W    W    W     ** FOR CURRENT PLACE, INDICATOR AND 3RD-LEVEL CODE:
136 W    W    W    W     DO WHILE provno+distno+ala+alb+alc=setdata
137 W    W    W    W    W
138 W    W    W    W    W     IF indic->indno$(ald->indic)
139 W    W    W    W    W    T     * flag to print totals after ends of loops:
140 W    W    W    W    W    T     store .T. to provtot,disttot,indictot
141 W    W    W    W    W    T     IF (rr>55.or.newprov) .and..not. pass1
142 W    W    W    W    W    T    T     eject
143 W    W    W    W    W    T    T     DO RPTLOCHD      && page header
144 W    W    W    W    W    T     ENDIF
145 W    W    W    W    W    T     store .F. to pass1,newprov
146 W    W    W    W    W    T     IF newdist
147 W    W    W    W    W    T    T     @ rr,1 say upper(trim(district->district))+
147 W    W    W    W    W    T    T     ' DISTRICT'
148 W    W    W    W    W    T    T     rr=rr+1
149 W    W    W    W    W    T    T     newdist=.F.
150 W    W    W    W    W    T     ENDIF
151 W    W    W    W    W    T     IF newindic
152 W    W    W    W    W    T    T     @ rr,3 say 'Objective: '+;
152 W    W    W    W    W    T    T      trim(indic->objective)+'; Indicator: '+;
152 W    W    W    W    W    T    T      indic->indicator
153 W    W    W    W    W    T    T     rr=rr+1
154 W    W    W    W    W    T    T     newindic=.F.
155 W    W    W    W    W    T     ENDIF
156 W    W    W    W    W    T     IF newalc        && print new 3rd-level code name
157 W    W    W    W    W    T    T     IF       (ala='5'.or.ala='6').and.'facility';
157 W    W    W    W    W    T    T     $(alb->albname)
158 W    W    W    W    W    T    T    T     facs=0
159 W    W    W    W    W    T    T    T     DO WHILE provno+distno+ala+alb+alc=setdata
160 W    W    W    W    W    T    T    T    W     IF 'salaries'$(ald->aldname)
161 W    W    W    W    W    T    T    T    W    T     facs=facs+qty
162 W    W    W    W    W    T    T    T    W     ENDIF
163 W    W    W    W    W    T    T    T    W     skip
164 W    W    W    W    W    T    T    T     ENDDO
165 W    W    W    W    W    T    T    T     seek setdata     && reposition pointer;
165 W    W    W    W    W    T    T    T      where it was
166 W    W    W    W    W    T    T    T     IF facs=1
167 W    W    W    W    W    T    T    T    T     @ rr,5 say trim(ala->alaname)+'/'+;
167 W    W    W    W    W    T    T    T    T      trim(alb->albname)+'/'+;
167 W    W    W    W    W    T    T    T    T     trim(alc->alcname)+ ' (1 facility)'
168 W    W    W    W    W    T    T    T     ELSE
169 W    W    W    W    W    T    T    T    F     @ rr,5 say trim(ala->alaname)+'/'+;
```

```
169 W   W   W   W   W   T   T   T   F     trim(alb->albname)+'/'+;
169 W   W   W   W   W   T   T   T   F    trim(alc->alcname)+' ('+;
169 W   W   W   W   W   T   T   T   F     ltrim(str(facs))+' facilities)'
170 W   W   W   W   W   T   T   T    ENDIF
171 W   W   W   W   W   T   T    ELSE
172 W   W   W   W   W   T   T   F    @ rr,5 say trim(ala->alaname)+'/'+;
172 W   W   W   W   W   T   T   F     trim(alb->albname)+'/'+alc->alcname
173 W   W   W   W   W   T   T    ENDIF   && (ala='5'.or.ala='6').and.';
173 W   W   W   W   W   T   T    facility'$(alb->albname)
174 W   W   W   W   W   T   T    rr=rr+1
175 W   W   W   W   W   T   T    newalc=.F.
176 W   W   W   W   W   T    ENDIF   && newalc
177 W   W   W   W   W   T
178 W   W   W   W   W   T    ** print a line
179 W   W   W   W   W   T    @ rr,7 say ald->aldname
180 W   W   W   W   W   T    @ rr,39 say conabbrev
181 W   W   W   W   W   T    @ rr,54 say qty pict '9999'
182 W   W   W   W   W   T    IF indic->indno<>'A'
183 W   W   W   W   W   T   T    @ rr,60 say ald->meas
184 W   W   W   W   W   T    ELSE
185 W   W   W   W   W   T   F    @ rr,60 say ald->priceunit
186 W   W   W   W   W   T    ENDIF
187 W   W   W   W   W   T    IF aidamt>0
188 W   W   W   W   W   T   T    @ rr,76 say aidamt pict '999999999'
189 W   W   W   W   W   T    ENDIF
190 W   W   W   W   W   T    IF transport>0
191 W   W   W   W   W   T   T    @ rr,87 say transport pict '9999999'
192 W   W   W   W   W   T    ENDIF
193 W   W   W   W   W   T    IF mafsbene>0
194 W   W   W   W   W   T   T    @ rr,96 say mafsbene pict '999999'
195 W   W   W   W   W   T    ENDIF
196 W   W   W   W   W   T    IF fafsbene>0
197 W   W   W   W   W   T   T    @ rr,103 say fafsbene pict '999999'
198 W   W   W   W   W   T    ENDIF
199 W   W   W   W   W   T    IF nafsbene>0
200 W   W   W   W   W   T   T    @ rr,110 say nafsbene pict '999999'
201 W   W   W   W   W   T    ENDIF
202 W   W   W   W   W   T    IF mafsbene+fafsbene+nafsbene>0
203 W   W   W   W   W   T   T    @ rr,117 say mafsbene+fafsbene+nafsbene pict;
203 W   W   W   W   W   T   T     '999999'
204 W   W   W   W   W   T    ENDIF
205 W   W   W   W   W   T    IF mafsbene+fafsbene+nafsbene>0
206 W   W   W   W   W   T   T    @ rr,125 say round((mafsbene+fafsbene)/;
206 W   W   W   W   W   T   T     (mafsbene+fafsbene+nafsbene)*100,0) pict ;
206 W   W   W   W   W   T   T    '999'+'%'
207 W   W   W   W   W   T    ENDIF
208 W   W   W   W   W   T    IF unsbene>0
209 W   W   W   W   W   T   T    @ rr,132 say unsbene pict '9999999'
210 W   W   W   W   W   T    ENDIF
211 W   W   W   W   W   T    IF indbene>0
212 W   W   W   W   W   T   T    @ rr,141 say indbene pict '9999999'
213 W   W   W   W   W   T    ENDIF
214 W   W   W   W   W   T    IF begin<>ctod('  /  /  ')
215 W   W   W   W   W   T   T    @ rr,151 say begin
216 W   W   W   W   W   T    ENDIF
217 W   W   W   W   W   T    IF end<>ctod('  /  /  ')
218 W   W   W   W   W   T   T    @ rr,161 say end
219 W   W   W   W   W   T    ENDIF
220 W   W   W   W   W   T    IF (aidamt+transport+mafsbene+fafsbene+nafsbene+;
220 W   W   W   W   W   T     unsbene+indbene)>0
221 W   W   W   W   W   T   T    DO CASE
```

```
222 W   W   W   W   W   T   T    CASE verstat=1  && positive
223 W   W   W   W   W   T   T   1    @ rr,174 say '1'
224 W   W   W   W   W   T   T   1    itstata=itstata+1
225 W   W   W   W   W   T   T    CASE verstat=0  && not yet verified
226 W   W   W   W   W   T   T   2    @ rr,179 say '1'
227 W   W   W   W   W   T   T   2    itstatb=itstatb+1
228 W   W   W   W   W   T   T    CASE verstat=-1 && negative
229 W   W   W   W   W   T   T   3    @ rr,184 say '1'
230 W   W   W   W   W   T   T   3    itstatc=itstatc+1
231 W   W   W   W   W   T   T    OTHERWISE       && partially monitored
232 W   W   W   W   W   T   T   0    @ rr,189 say '1'
233 W   W   W   W   W   T   T   0    itstatd=itstatd+1
234 W   W   W   W   W   T   T    ENDCASE
235 W   W   W   W   W   T    ENDIF
236 W   W   W   W   W   T    IF verdate<>ctod('  /  /  ')
237 W   W   W   W   W   T   T    @ rr,192 say verdate
238 W   W   W   W   W   T    ENDIF
239 W   W   W   W   W   T    rr=rr+1
240 W   W   W   W   W   T    ** increment indicator totals
241 W   W   W   W   W   T    itaidamt=itaidamt+aidamt
242 W   W   W   W   W   T    ittranspor=ittranspor+transport
243 W   W   W   W   W   T    itmafsbene=itmafsbene+mafsbene
244 W   W   W   W   W   T    itfafsbene=itfafsbene+fafsbene
245 W   W   W   W   W   T    itnafsbene=itnafsbene+nafsbene
246 W   W   W   W   W   T    itunsbene=itunsbene+unsbene
247 W   W   W   W   W   T    itindbene=itindbene+indbene
248 W   W   W   W   W    ENDIF   && indic->indno$(ald->indic)
249 W   W   W   W   W    skip
250 W   W   W   W    ENDDO   && provno+distno+ala+alb+alc=setdata
251 W   W   W   W    select alc
252 W   W   W   W    skip
253 W   W   W    ENDDO   && .not. eof() in alc.dbf
254 W   W   W
255 W   W   W    IF indictot     && print indicator totals
256 W   W   W   T    ** print indicator totals:
257 W   W   W   T    @ rr,76 say repl('-',83)
258 W   W   W   T    @ rr,172 say repl('-',18)
259 W   W   W   T    rr=rr+1
260 W   W   W   T    @ rr,5 say 'Indicator Totals'
261 W   W   W   T    @ rr,76 say itaidamt pict '999999999'
262 W   W   W   T    @ rr,87 say ittranspor pict '9999999'
263 W   W   W   T    @ rr,96 say itmafsbene pict '999999'
264 W   W   W   T    @ rr,103 say itfafsbene pict '999999'
265 W   W   W   T    @ rr,110 say itnafsbene pict '999999'
266 W   W   W   T    @ rr,117 say itmafsbene+itfafsbene+itnafsbene pict ;
266 W   W   W   T    '999999'
267 W   W   W   T    IF itmafsbene+itfafsbene+itnafsbene>0
268 W   W   W   T   T    @ rr,125 say round((itmafsbene+itfafsbene)/;
268 W   W   W   T   T     (itmafsbene+itfafsbene+itnafsbene)*100,0) pict ;
268 W   W   W   T   T    '999'+'%'
269 W   W   W   T    ENDIF
270 W   W   W   T    @ rr,132 say itunsbene pict '9999999'
271 W   W   W   T    @ rr,141 say itindbene pict '9999999'
272 W   W   W   T    @ rr,172 say itstata pict '999'
273 W   W   W   T    @ rr,177 say itstatb pict '999'
274 W   W   W   T    @ rr,182 say itstatc pict '999'
275 W   W   W   T    @ rr,187 say itstatd pict '999'
276 W   W   W   T    rr=rr+2
277 W   W   W   T    ** add indicator totals to district totals
278 W   W   W   T    dtaidamt=dtaidamt+itaidamt
279 W   W   W   T    dttranspor=dttranspor+ittranspor
```

```
280 W    W    W    T     dtmafsbene=dtmafsbene+itmafsbene
281 W    W    W    T     dtfafsbene=dtfafsbene+itfafsbene
282 W    W    W    T     dtnafsbene=dtnafsbene+itnafsbene
283 W    W    W    T     dtunsbene=dtunsbene+itunsbene
284 W    W    W    T     dtindbene=dtindbene+itindbene
285 W    W    W    T     dtstata=dtstata+itstata
286 W    W    W    T     dtstatb=dtstatb+itstatb
287 W    W    W    T     dtstatc=dtstatc+itstatc
288 W    W    W    T     dtstatd=dtstatd+itstatd
289 W    W    W    T     ** reinitialize indicator accumulators:
290 W    W    W    T     store 0 to itaidamt,ittranspor,itmafsbene,itfafsbene,;
290 W    W    W    T      itnafsbene,itunsbene,itindbene,itstata,itstatb,itstatc,;
290 W    W    W    T     itstatd
291 W    W    W    T     indictot=.F.
292 W    W    W     ENDIF && indictot
293 W    W    W     select indic
294 W    W    W     skip
295 W    W     ENDDO   && .not. eof() for indic
296 W    W
297 W    W     IF disttot       && print district totals
298 W    W    T     @ rr,76 say repl('-',83)
299 W    W    T     @ rr,172 say repl('-',18)
300 W    W    T     rr=rr+1
301 W    W    T     @ rr,3 say 'DISTRICT TOTALS'
302 W    W    T     @ rr,76 say dtaidamt pict '999999999'
303 W    W    T     @ rr,87 say dttranspor pict '9999999'
304 W    W    T     @ rr,96 say dtmafsbene pict '999999'
305 W    W    T     @ rr,103 say dtfafsbene pict '999999'
306 W    W    T     @ rr,110 say dtnafsbene pict '999999'
307 W    W    T     @ rr,117 say dtmafsbene+dtfafsbene+dtnafsbene pict '999999'
308 W    W    T     IF dtmafsbene+dtfafsbene+dtnafsbene>0
309 W    W    T    T     @ rr,125 say round((dtmafsbene+dtfafsbene)/ (dtmafsbene+;
309 W    W    T    T     dtfafsbene+dtnafsbene)*100,0) pict '999'+'%'
310 W    W    T     ENDIF
311 W    W    T     @ rr,132 say dtunsbene pict '9999999'
312 W    W    T     @ rr,141 say dtindbene pict '9999999'
313 W    W    T     @ rr,172 say dtstata pict '999'
314 W    W    T     @ rr,177 say dtstatb pict '999'
315 W    W    T     @ rr,182 say dtstatc pict '999'
316 W    W    T     @ rr,187 say dtstatd pict '999'
317 W    W    T     rr=rr+1
318 W    W    T     ** add district totals to province totals:
319 W    W    T     ptaidamt=ptaidamt+dtaidamt
320 W    W    T     pttranspor=pttranspor+dttranspor
321 W    W    T     ptmafsbene=ptmafsbene+dtmafsbene
322 W    W    T     ptfafsbene=ptfafsbene+dtfafsbene
323 W    W    T     ptnafsbene=ptnafsbene+dtnafsbene
324 W    W    T     ptunsbene=ptunsbene+dtunsbene
325 W    W    T     ptindbene=ptindbene+dtindbene
326 W    W    T     ptstata=ptstata+dtstata
327 W    W    T     ptstatb=ptstatb+dtstatb
328 W    W    T     ptstatc=ptstatc+dtstatc
329 W    W    T     ptstatd=ptstatd+dtstatd
330 W    W    T     ** reinitialize province accumulators:
331 W    W    T     store 0 to dtaidamt,dttranspor,dtmafsbene,dtfafsbene,;
331 W    W    T      dtnafsbene,dtunsbene,dtindbene,dtstata,dtstatb,dtstatc,;
331 W    W    T     dtstatd
332 W    W    T     disttot=.F.
333 W    W     ENDIF && disttot
334 W    W     select district
335 W    W     skip
```

```
336 W     ENDDO   && .not. eof() for district
337 W     IF provtot       && print province totals
338 W   T     @ rr,76 say repl('-',83)
339 W   T     @ rr,172 say repl('-',18)
340 W   T     rr=rr+1
341 W   T     IF province->provno<>'99'
342 W   T   T     @ rr,3 say 'PROVINCE TOTALS'
343 W   T     ELSE
344 W   T   F     @ rr,3 say 'PAKISTAN TOTALS'
345 W   T     ENDIF
346 W   T     @ rr,76 say ptaidamt pict '999999999'
347 W   T     @ rr,87 say pttranspor pict '9999999'
348 W   T     @ rr,96 say ptmafsbene pict '999999'
349 W   T     @ rr,103 say ptfafsbene pict '999999'
350 W   T     @ rr,110 say ptnafsbene pict '999999'
351 W   T     @ rr,117 say ptmafsbene+ptfafsbene+ptnafsbene pict '999999'
352 W   T     IF ptmafsbene+ptfafsbene+ptnafsbene>0
353 W   T   T     @ rr,125 say round((ptmafsbene+ptfafsbene)/ (ptmafsbene+;
353 W   T   T     ptfafsbene+ptnafsbene)*100,0) pict '999'+'%'
354 W   T     ENDIF
355 W   T     @ rr,132 say ptunsbene pict '9999999'
356 W   T     @ rr,141 say ptindbene pict '9999999'
357 W   T     @ rr,172 say ptstata pict '999'
358 W   T     @ rr,177 say ptstatb pict '999'
359 W   T     @ rr,182 say ptstatc pict '999'
360 W   T     @ rr,187 say ptstatd pict '999'
361 W   T     rr=rr+1
362 W   T     provtot=.F.
363 W     ENDIF && disttot
364 W     select province
365 W     skip
366 ENDDO   && .not. eof() in province.dbf
367 eject
368 eject
369 set device to screen
370 close all
371 return
372
```

++ End of Program RPTLOC ++ DOC found no errors.

VARIABLES: LOCAL CONCORDANCE FOR RPTLOC

Variable	Statement number in listing (modified variables indicated by '=')						
AIDAMT:	187	188	220	241			
ALA:	22	25	49	124	136	157	159
ALANAME:	167	169	172				
ALB:	25	49	124	136	159		
ALBNAME:	157	167	169	172			
ALC:	49	124	136	159			
ALCNAME:	167	169	172				
ALD:	49						
ALDNAME:	160	179					
answer:	=36	=40	45				
BEGIN:	214	215					
CONABBREV:	180						
DISTNO:	83	112	124	136	159		
DISTRICT:	147						
disttot:	=56	=140	297	=332			
DONATED:	103	105					
DPI:	107						

VARIABLES: LOCAL CONCORDANCE FOR RPTLOC (continued)

Variable	Statement number in listing (modified variables indicated by '=')									
dtaidamt:	=54	=278	302	319	=331					
dtfafsbene:	=54	=281	305	307	308	309	322	=331		
dtindbene:	=54	=284	312	325	=331					
dtmafsbene:	=54	=280	304	307	308	309	321	=331		
dtnafsbene:	=54	=282	306	307	308	309	323	=331		
dtstata:	=54	=285	313	326	=331					
dtstatb:	=54	=286	314	327	=331					
dtstatc:	=54	=287	315	328	=331					
dtstatd:	=54	=288	316	329	=331					
dttranspor:	=54	=279	303	320	=331					
dtunsbene:	=54	=283	311	324	=331					
END:	217	218								
facs:	=158	=161	166	169						
FAFSBENE:	196	197	202	203	205	206	220	244		
INDBENE:	211	212	220	247						
INDIC:	138									
INDICATOR:	152									
indictot:	=56	=140	255	=291						
INDNO:	100	102	104	106	138	182				
itaidamt:	=53	=241	261	278	=290					
itfafsbene:	=53	=244	264	266	267	268	281	=290		
itindbene:	=53	=247	271	284	=290					
itmafsbene:	=53	=243	263	266	267	268	280	=290		
itnafsbene:	=53	=245	265	266	267	268	282	=290		
itstata:	=53	=224	272	285	=290					
itstatb:	=53	=227	273	286	=290					
itstatc:	=53	=230	274	287	=290					
itstatd:	=53	=233	275	288	=290					
ittranspor:	=53	=242	262	279	=290					
itunsbene:	=53	=246	270	283	=290					
MAFSBENE:	193	194	202	203	205	206	220	243		
MEAS:	183									
NAFSBENE:	199	200	202	203	205	206	220	245		
newalc:	=132	156	=175							
newdist:	=89	146	=149							
newindic:	=118	151	=154							
newprov:	=76	141	=145							
OBJECTIVE:	152									
pass1:	=66	90	141	=145						
pp:	=59									
PRICE:	101	103	105							
PRICEUNIT:	185									
print:	62									
PROVNO:	10	70	80	81	83	112	124	136	159	341
provtot:	=56	=140	337	=362						
ptaidamt:	=55	=319	346							
ptfafsbene:	=55	=322	349	351	352	353				
ptindbene:	=55	=325	356							
ptmafsbene:	=55	=321	348	351	352	353				
ptnafsbene:	=55	=323	350	351	352	353				
ptstata:	=55	=326	357							
ptstatb:	=55	=327	358							
ptstatc:	=55	=328	359							
ptstatd:	=55	=329	360							
pttranspor:	=55	=320	347							
ptunsbene:	=55	=324	355							
QTR:	101	103	105	107	109					

VARIABLES: LOCAL CONCORDANCE FOR RPTLOC (continued)

ariable	Statement number in listing (modified variables indicated by '=')										
trset:	=31	=33	101	103	105	107	109				
TY:	161	181									
r:	=58	141	147	=148	152	=153	167	169	172	=174	179
	180	181	183	185	188	191	194	197	200	203	206
	209	212	215	218	223	226	229	232	237	=239	257
	258	=259	260	261	262	263	264	265	266	268	270
	271	272	273	274	275	=276	298	299	=300	301	302
	303	304	305	306	307	309	311	312	313	314	315
	316	=317	338	339	=340	342	344	346	347	348	349
	350	351	353	355	356	357	358	359	360	=361	
screen:	369										
setdata:	=124	126	136	159	165						
TRANSPORT:	190	191	220	242							
JNSBENE:	208	209	220	246							
VERDATE:	236	237									
VERSTAT:	222	225	228								
YEAR:	101	103	105	107	109						
yearset:	=32	=34	101	103	105	107	109				

```
 1 * util.prg - utilities for rep.prg
 2 * ver 1.0 Ron Barney 23 November 1990
 3
 4 DO WHILE .T.
 5 W     clear
 6 W     @  1,20 say 'USAID/Representative for Afghanistan/DC&A'
 7 W     @  2,29 say 'PERFORMANCE DATA SYSTEM'
 8 W     @  3,1 to 3,78
 9 W     @  5,20 say 'Utilities:'
10 W     @  7,20 say '1 - Download repdata.dbf to drive B:'
11 W     @  9,20 say '2 - ZAP repdata.dbf from hard drive'
12 W     @ 11,20 say '3 - Append b:repdata.dbf from diskette to'
13 W     @ 12,20 say '     main database.'
14 W     @ 14,20 say '4 - Rebuild indexes'
15 W     answer='0'
16 W     @ 17,20 say 'Selection: ' get answer pict '!'
17 W     read
18 W     DO CASE
19 W     CASE answer='0'
20 <========EXIT     && from current loop back to main menu
21 W     CASE answer='1' && download to b:
22 W   2     use repdata
23 W   2     ** This should be rewritten to automatically split the
24 W   2     ** file and dump the parts on diskette:
25 W   2     DO WHILE .T.
26 W   2   W     IF (reccount()*recsize()+1000)>=729000
27 W   2   W   T     @ 19,0 clear
28 W   2   W   T     @ 19,20 say 'Sorry, Repdata.dbf is too big to download'
29 W   2   W   T     @ 20,20 say 'to a 730112-byte diskette.  Exit and use'
30 W   2   W   T     @ 21,20 say 'dBase to split the file...'
31 W   2   W   T     @ 22,0
32 W   2   W   T     wait
33 W   2<===========EXIT
34 W   2   W     ENDIF
35 W   2   W     @ 19,0 clear
36 W   2   W     @ 19,20 say 'Insert a formatted, 730112-byte diskette'
37 W   2   W     wait space(20)+'in drive B: and press any key to continue...'
38 W   2   W     run copy repdata.dbf b:/v
39 W   2   W     @ 19,0 clear
40 W   2   W     answer=' '
41 W   2   W     @ 19,20 say 'You should make at least two copies.'
42 W   2   W     @ 20,20 say 'Do you want to make another copy?' get answer;
42 W   2   W      pict 'Y'
43 W   2   W     read
44 W   2   W     IF answer='N'
45 W   2<===========EXIT
46 W   2   W     ENDIF
47 W   2     ENDDO
48 W   2
49 W     CASE answer='2' && ZAP main database.
50 W   3     answer=' '
51 W   3     @ 19,20 SAY 'Have you copied REPDATA.DBF to two'
52 W   3     @ 20,20 say 'diskettes' get answer pict 'Y'
53 W   3     read
54 W   3     password=space(1)
55 W   3     @ 21,0
56 W   3     accept space(20)+'Please enter password --> ' to password
57 W   3     IF answer='Y' .and. upper(password)='USAID'
58 W   3   T     use repdata index repdata
59 W   3   T     zap
60 W   3   T     reindex
```

```
61 W    3    T     use
62 W    3     ELSE
63 W    3    F     ? chr(7)
64 W    3    F     @ 18,0 clear
65 W    3    F     wait space(20)+'Zap not performed.'
66 W    3     ENDIF
67 W     CASE answer='3' && upload from diskette
68 W    4     @ 19,20 say 'Uploading...'
69 W    4     use repdata index repdata
70 W    4     append from b:repdata.dbf
71 W    4     ? chr(7)
72 W     CASE answer='4'
73 W    5     answer='N'
74 W    5     @ 19,20 say 'Reindexing can take several minutes.'
75 W    5     @ 20,20 say 'Are you sure you want to go ahead?' get answer pict;
75 W    5      'Y'
76 W    5     read
77 W    5     IF answer='Y'
78 W    5    T     DO REBUILD
79 W    5     ENDIF
80 W     OTHERWISE
81 W    0     ? chr(7)
82 W     ENDCASE
83 ENDDO
84
```

++ End of Program UTIL ++ DOC found no errors.

VARIABLES: LOCAL CONCORDANCE FOR UTIL

Variable	Statement number in listing (modified variables indicated by '=')										
answer:	=15	=16	19	21	=40	=42	44	49	=50	=52	57
	67	72	=73	=75	77						
password:	=54	=56	57								

```
1 * location.prg - identify location in Afghanistan
2 * ver 2 Ron Barney 23 October 1990
3 * General logic:  province, district and village are to be chosen in order.
4 * Subroutines for making these choices are controlled within the main loop
5 * by variables Pchoose, Dchoose and Vchoose.  Exit from any of the
5 *  subroutines
6 * can determine new values for the three variables.  If remaining
6 *  variable(s)
7 * .F., control goes back to the top of the loop, the current defaults
7 *  display,
8 * and appropriate subroutines are exercised.  Exit only if village choice
9 * accepted.
10
11 * Clipper function lastkey() has been used to test for the ESC key
12 * (ASCII 27) both here and in calling program ENTRY.  IF ESC is read from
13 * the keyboard buffer, the RETURN command is always issued.  While it is
14 * normally better structured programming to handle this kind of thing
15 * differently, there are are so many reads involved that this exceptional
16 * handling seems to make the code more clear, rather than the reverse.
17
18 ****************************
19 *select 4
20 *use D:\gaz\province index D:\gaz\provno
21 *select 5
22 *use D:\gaz\district index D:\gaz\pdistrict, D:\gaz\pdistno
23 *select 6
24 *use D:\gaz\place index D:\gaz\pdplace, D:\gaz\pdplaceno
25 *select 7
26 *use D:\gaz\newplace
27 * default variables from calling program:
28 *nprov=space(2)
29 *ndist=space(4)
30 *nplace=space(3)
31 *cprov=space(25)          && label for province
32 *cdist=space(19)          && label for district
33 *cplace=space(40)         && label for place
34 ******************************************
35
36 * ccs=20          && column where < sign will appear.  Assigned in calling
36 *  program
37 cct=ccs+2         && column where location text will begin
38 rrp=2             && row for province
39 rrd=3             && row for district
40 rrv=4             && row for village
41
42 store .T. to Pchoose, Dchoose, Vchoose
43 DO WHILE .T.
44 W     @ 0,ccs CLEAR TO 24,79
45 W     * DISPLAY DEFAULT LOCATION FOR CONFIRMATION
46 W     IF nprov<>space(2).and.ndist<>space(4).and.nplace<>space(3)
47 W    T     store .F. to Pchoose, Dchoose
48 W    T     Vchoose=.T.
49 W     ENDIF for IF nprov<>space(2)
50 W
51 W     IF Pchoose
52 W    T     ** province **
53 W    T     select province
54 W    T     DO WHILE .T.
55 W    T    W     @ 0,ccs clear to 24,79
56 W    T    W     IF nprov<>space(2)       && default value has been set up
57 W    T    W    T     rr=rrp  && row for display of confirmation message
```

```
58 W   T   W   T    @ rrp,ccs-3 say nprov
59 W   T   W   T    @ rrp,ccs say '< '+cprov
60 W   T   W   T    IF nprov='00'
61 W   T   W   T   T    @ rrd,ccs-5 say ndist+' < '+cdist
62 W   T   W   T   T    @ rrv,ccs-4 say nplace+' < '+cplace
63 W   T   W   T   T    rr=rrv
64 W   T   W   T    ENDIF
65 W   T   W   T    answer=' '
66 W   T   W   T    @ rr+2,cct say 'Enter blank to confirm,'
67 W   T   W   T    @ rr+3,cct say 'any other key to reselect -->' get answer
68 W   T   W   T    read
69 W   T   W   T    DO CASE
70 W   T   W   T    CASE lastkey()=27
71 <---------------------return
72 W   T   W   T    CASE answer=' '.and.nprov='00'
73 W   T   W   T   2    store .F. to Dchoose, Vchoose
74 W   T<===============EXIT
75 W   T   W   T    CASE answer=' '
76 W   T   W   T   3    store .T. to Dchoose, Vchoose
77 W   T<===============EXIT
78 W   T   W   T    OTHERWISE
79 W   T   W   T   0    nprov=space(2)
80 W   T   W   T   0    ndist=space(4)
81 W   T   W   T   0    nplace=space(3)
82 W   T   W   T   0    cprov=space(25) && label for province
83 W   T   W   T   0    cdist=space(19) && label for district
84 W   T   W   T   0    cplace=space(40)        && label for place
85 W   T   W   T   0    @ rrd,ccs-5 say ndist
86 W   T   W   T   0    @ rrv,ccs-4 say nplace
87 W   T   W   T   0    go top
88 W   T   ^============loop
89 W   T   W   T    ENDCASE
90 W   T   W    ELSE    && no default value in force
91 W   T   W   F    @ rrp,ccs say '<'
92 W   T   W   F    @  1,cct say '11 Badakshan                10 Konar';
92 W   T   W   F    + ' (Kunarah)'
93 W   T   W   F    @  2,cct say '19 Badghis (Badghisat)      14 Kunduz'
94 W   T   W   F    @  3,cct say '13 Baghlan                  09 Laghman'
95 W   T   W   F    @  4,cct say '16 Balkh                    05 Logar'
96 W   T   W   F    @  5,cct say '28 Bamyan                   08 Nangarhar'
97 W   T   W   F    @  6,cct say '21 Farah                    22 Nimroz'
98 W   T   W   F    @  7,cct say '18 Faryab                   26 Oruzgan'
99 W   T   W   F    @  8,cct say '06 Ghazni                   29 Pakteka'
100 W  T   W   F    @  9,cct say '27 Ghor (Ghawrat)           07 Paktya'
101 W  T   W   F    @ 10,cct say '23 Helmand                  03 Parwan'
102 W  T   W   F    @ 11,cct say '20 Herat                    15 Samangan'
103 W  T   W   F    @ 12,cct say '17 Jawzjan                  30 Sari Pul'
104 W  T   W   F    @ 13,cct say '01 Kabul                    12 Takhar'
105 W  T   W   F    @ 14,cct say '24 Kandahar (Quandahar)     04 Wardak';
105 W  T   W   F    + ' (Maydan)'
106 W  T   W   F    @ 15,cct say '02 Kapisa                   25 Zabul' '
107 W  T   W   F    @ 17,cct say '  99 PAKISTAN              00 UNSPECIFIED'
108 W  T   W   F    numberprov=0
109 W  T   W   F    @ rrp,ccs-3 get numberprov pict '99'
110 W  T   W   F    read
111 W  T   W   F    nprov=subst(str(100+numberprov,3),2)
112 W  T   W    ENDIF
113 W  T   W    IF lastkey()=27
114 <-----------------return
115 W  T   W    ENDIF
116 W  T   W    seek nprov
```

```
117 W    T    W     IF .not. found()
118 W    T    W    T     nprov=space(2)
119 W    T    W    T     ? chr(7)
120 W    T    ^=========loop
121 W    T    W     ELSE
122 W    T    W    F     cprov=province
123 W    T    W    F     IF nprov='00'.or.nprov='99'
124 W    T    W    F    T     IF nprov='00'
125 W    T    W    F    T    T     cprov='Unspecified'
126 W    T    W    F    T     ELSE
127 W    T    W    F    T    F     cprov='Pakistan'
128 W    T    W    F    T     ENDIF
129 W    T    W    F    T     ndist='0000'
130 W    T    W    F    T     cdist='Unspecified'
131 W    T    W    F    T     nplace='000'
132 W    T    W    F    T     cplace='Unspecified'
133 W    T    W    F     ENDIF
134 W    T    W     ENDIF
135 W    T     ENDDO
136 W     ENDIF for Pchoose=.T.
137 W
138 W     IF Dchoose       && district;
138 W      %*****************************************
139 W    T     select district
140 W    T     set order to 1
141 W    T     seek nprov
142 W    T     DO WHILE .T.
143 W    T    W     @ 0,ccs clear to 24,79
144 W    T    W     IF ndist<>space(4) && default value not in force: display;
144 W    T    W      values and get one
145 W    T    W    T     rr=14    && row for display of confirmation message
146 W    T    W    T     @ rrp,ccs-3 say nprov+' < '+cprov
147 W    T    W    T     @ rrd,ccs-5 say ndist+' < '+cdist
148 W    T    W    T     IF ndist='0000'
149 W    T    W    T    T     @ rrv,ccs-4 say nplace+' < '+cplace
150 W    T    W    T    T     rr=15
151 W    T    W    T     ENDIF
152 W    T    W    T     answer=' '
153 W    T    W    T     @ rrv+2,cct say 'Enter blank to confirm,'
154 W    T    W    T     @ rrv+3,cct say 'P to reselect province,'
155 W    T    W    T     @ rrv+4,cct say 'D to reselect district-->' get answer;
155 W    T    W    T      pict '!'
156 W    T    W    T     read
157 W    T    W    T     DO CASE
158 W    T    W    T     CASE lastkey()=27
159 <----------------------return
160 W    T    W    T     CASE answer=' '.and. ndist='0000'
161 W    T    W    T    2     Vchoose=.F.
162 W    T<===============EXIT && choice of district
163 W    T    W    T     CASE answer=' '
164 W    T    W    T    3     Vchoose=.T.
165 W    T<===============EXIT && choice of district
166 W    T    W    T     CASE answer='P'
167 W    T    W    T    4     nprov=space(2)
168 W    T    W    T    4     ndist=space(4)
169 W    T    W    T    4     nplace=space(3)
170 W    T    W    T    4     cprov=space(25) && label for province
171 W    T    W    T    4     cdist=space(19) && label for district
172 W    T    W    T    4     cplace=space(40)         && label for place
173 W    T    W    T    4     @ rrp,ccs-3 say space(2)
174 W    T    W    T    4     @ rrd,ccs-5 say space(4)
```

```
75 W    T    W    T    4     @ rrv,ccs-4 say space(3)
76 W    T    W    T    4     store .T. to Pchoose
77 W    T    W    T    4     store .F. to Dchoose, Vchoose
78 W    T<===============EXIT && choice of district
79 W    T    W    T     OTHERWISE        && 'D' or other response
80 W    T    W    T    0     ndist=space(4)
81 W    T    W    T    0     nplace=space(3)
82 W    T    W    T    0     cdist=space(19) && label for district
83 W    T    W    T    0     cplace=space(40)         && label for place
84 W    T    W    T    0     @ rrd,ccs-5 say ndist
85 W    T    W    T    0     @ rrv,ccs-4 say nplace
86 W    T    W    T    0     seek nprov
87 W    T    ^============loop
88 W    T    W    T     ENDCASE
89 W    T    W    ·ELSE
90 W    T    W    F     longlist=.F.     && controls display of scroll message
91 W    T    W    F     DO WHILE .T.
92 W    T    W    F    W     @ 0,ccs clear to 24,79
93 W    T    W    F    W     @ rrd,ccs say '<'
94 W    T    W    F    W     rr=1     && row counter
95 W    T    W    F    W     DO WHILE rr<22 .and.provno=nprov .and..not.eof()
96 W    T    W    F    W    W     @ rr,cct say distno+space(2)+district
97 W    T    W    F    W    W     rr=rr+1
98 W    T    W    F    W    W     skip
99 W    T    W    F    W     ENDDO
200 W   T    W    F    W     IF provno<>nprov.or.eof()
201 W   T    W    F    W    T     @ rr,cct say '*** End of list ***'
202 W   T    W    F    W     ELSE
203 W   T    W    F    W    F     longlist=.T.
204 W   T    W    F    W     ENDIF
205 W   T    W    F    W     IF longlist
206 W   T    W    F    W    T     @ 23,cct say 'Enter number to select, '
207 W   T    W    F    W    T     @ 24,cct say chr(25)+' to see more districts.'
208 W   T    W    F    W     ENDIF
209 W   T    W    F    W     numberdist=0
210 W   T    W    F    W     readexit(.T.)    && clipper command
211 W   T    W    F    W     @ rrd,ccs-5 GET numberdist pict '9999'
212 W   T    W    F    W     read
213 W   T    W    F    W     readexit(.F.)    && clipper command
214 W   T    W    F    W     IF lastkey()=27
215 <-------------------------return
216 W   T    W    F    W     ENDIF
217 W   T    W    F    W     IF lastkey()=24 && clipper scroll
218 W   T    W    F    W    T     *                                    IF
218 W   T    W    F    W    T     *  readkey()=5 .or. readkey()=261    && doase
218 W   T    W    F    W    T     *  scroll
219 W   T    W    F    W    T     IF (provno<>nprov).or.eof()
220 W   T    W    F    W    T    T     seek nprov       && always found
221 W   T    W    F    W    T    T     * else continue from where you are.
222 W   T    W    F    W    T     ENDIF
223 W   T    W    F    ^========loop
224 W   T    W    F    W     ELSE
225 W   T    W    F    W    F     ndist=subst(str(10000+numberdist,5),2)
226 W   T    W    F    W     ENDIF
227 W   T    W    F<=======EXIT
228 W   T    W    F     ENDDO
229 W   T    W     ENDIF    && choose ndist
230 W   T    W     set order to 2
231 W   T    W     seek nprov+ndist
232 W   T    W     IF .not. found()
233 W   T    W    T     ? chr(7)
```

```
234 W    T    W    T     ndist=space(4)
235 W    T    W    T     set order to 1
236 W    T    W    T     seek nprov
237 W    T    ^=========loop
238 W    T    W     ELSE
239 W    T    W    F     set order to 1
240 W    T    W    F     cdist=district
241 W    T    W    F     IF ndist='0000'
242 W    T    W    F    T     nplace='000'
243 W    T    W    F    T     cdist='Unspecified'
244 W    T    W    F    T     cplace='Unspecified'
245 W    T    W    F     ENDIF
246 W    T    W     ENDIF
247 W    T     ENDDO
248 W     ENDIF for Dchoose=.T.
249 W
250 W     IF Vchoose       && place;
250 W      ***********************************************
251 W    T     select place
252 W    T     seek nprov+ndist
253 W    T     DO WHILE .T.
254 W    T    W     @ 0,ccs clear to 24,79
255 W    T    W     IF nplace<>space(3)
256 W    T    W    T     answer=' '
257 W    T    W    T     @ rrp,ccs-3 say nprov+' < '+cprov
258 W    T    W    T     @ rrd,ccs-5 say ndist+' < '+cdist
259 W    T    W    T     @ rrv,ccs-4 say nplace+' < '+cplace
260 W    T    W    T     @ rrv+2,cct say 'Enter blank to confirm,'
261 W    T    W    T     @ rrv+3,cct say 'P to reselect province,'
262 W    T    W    T     @ rrv+4,cct say 'D to reselect district,'
263 W    T    W    T     @ rrv+5,cct say 'V to reselect village -->' get answer;
263 W    T    W    T      pict '!'
264 W    T    W    T     read
265 W    T    W    T     DO CASE
266 W    T    W    T     CASE lastkey()=27
267 <----------------------return
268 W    T    W    T     CASE answer=' '
269 W    T<================EXIT && choice of village
270 W    T    W    T     CASE answer='P'
271 W    T    W    T    3     nprov=space(2)
272 W    T    W    T    3     ndist=space(4)
273 W    T    W    T    3     nplace=space(3)
274 W    T    W    T    3     cprov=space(25) && label for province
275 W    T    W    T    3     cdist=space(19) && label for district
276 W    T    W    T    3     cplace=space(40)         && label for place
277 W    T    W    T    3     @ rrp,ccs-3 say space(2)
278 W    T    W    T    3     @ rrd,ccs-5 say space(4)
279 W    T    W    T    3     @ rrv,ccs-4 say space(3)
280 W    T    W    T    3     store .T. to Pchoose, Dchoose, Vchoose
281 W    T<================EXIT && choice of village
282 W    T    W    T     CASE answer='D'
283 W    T    W    T    4     ndist=space(4)
284 W    T    W    T    4     nplace=space(3)
285 W    T    W    T    4     cdist=space(19) && label for district
286 W    T    W    T    4     cplace=space(40)         && label for place
287 W    T    W    T    4     @ rrd,ccs-5 say space(4)
288 W    T    W    T    4     @ rrv,ccs-4 say space(3)
289 W    T    W    T    4     Pchoose=.F.
290 W    T    W    T    4     store .T. to Dchoose, Vchoose
291 W    T<================EXIT     && choice of village
292 W    T    W    T     OTHERWISE
```

```
293 W   T   W   T   O    nplace=space(3)
294 W   T   W   T   O    cplace=space(40)          && label for place
295 W   T   W   T   O    @ rrv,ccs-4 say nplace
296 W   T   W   T   O    seek nprov+ndist
297 W   T   ^=============loop
298 W   T   W   T     ENDCASE
299 W   T   W    ELSE    && nplace=space(3)
300 W   T   W   F     longlist=.F.     && controls display of scroll message
301 W   T   W   F     DO WHILE .T.
302 W   T   W   F   W    @ 0,ccs clear to 24,79
303 W   T   W   F   W    @ rrv,ccs say '<'
304 W   T   W   F   W    rr=1     && row counter
305 W   T   W   F   W    DO WHILE rr<21 .and.provno+distno=nprov+ndist;
305 W   T   W   F   W     .and..not.eof()
306 W   T   W   F   W   W    @ rr,cct say pl. eno+space(1)+place
307 W   T   W   F   W   W    rr=rr+1
308 W   T   W   F   W   W    skip
309 W   T   W   F   W    ENDDO
310 W   T   W   F   W    IF provno+distno<>nprov+ndist.or.eof()
311 W   T   W   F   W   T    @ rr,cct say '*** End of list ***'
312 W   T   W   F   W    ELSE
313 W   T   W   F   W   F    longlist=.T.
314 W   T   W   F   W    ENDIF
315 W   T   W   F   W    IF longlist
316 W   T   W   F   W   T    @ 22,cct say 'Enter number to select, letter for'
317 W   T   W   F   W   T    @ 23,cct say 'lookup, '+chr(25)+' to see more;
317 W   T   W   F   W   T     villages.'
318 W   T   W   F   W    ENDIF
319 W   T   W   F   W    nplace2=space(3)
320 W   T   W   F   W    readexit(.T.)    && clipper command
321 W   T   W   F   W    @ rrv,ccs-4 get nplace2 pict '@! X99'
322 W   T   W   F   W    read
323 W   T   W   F   W    readexit(.F.)    && clipper command
324 W   T   W   F   W    DO CASE
325 W   T   W   F   W    CASE lastkey()=27
326 <-------------------------------return
327 W   T   W   F   W   1    *
327 W   T   W   F   W   1    *  CASE readkey()=5 .or. readkey()=261      &&
327 W   T   W   F   W   1    *  down arrow (dbase)
328 W   T   W   F   W    CASE lastkey()=24        && down arrow entered;
328 W   T   W   F   W     (clipper)
329 W   T   W   F   W   2    IF (provno+distno<>nprov+ndist).or.eof()
330 W   T   W   F   W   2   T    seek nprov+ndist          && always found
331 W   T   W   F   W   2   T    * else continue from where you are.
332 W   T   W   F   W   2    ENDIF
333 W   T   W   F   ^========loop
334 W   T   W   F   W    CASE subst(nplace2,1,1)='*'
335 W   T   W   F   W   3    @ 0,ccs clear to 24,79
336 W   T   W   F   W   3    nplace='***'
337 W   T   W   F   W   3    cplace=space(40)
338 W   T   W   F   W   3    @ rrv,ccs-4 say '*** < Not listed in database'
339 W   T   W   F   W   3    answer=' '
340 W   T   W   F   W   3    @ rrv+2,cct say 'Enter blank to confirm,'
341 W   T   W   F   W   3    @ rrv+3,cct say 'any other key to reselect -->';
341 W   T   W   F   W   3     get answer
342 W   T   W   F   W   3    read
343 W   T   W   F   W   3    IF lastkey()=27
344 <-------------------------------return
345 W   T   W   F   W   3    ENDIF
346 W   T   W   F   W   3    IF answer=' '
347 W   T   W   F   W   3   T    cplace=space(40)
```

```
348 W   T   W   F   W   3   T    @ 0,ccs clear to 24,79
349 W   T   W   F   W   3   T    DO WHILE len(trim(cplace))=0
350 W   T   W   F   W   3   T   W    @ rrv,ccs-4 say '*** <' get cplace pict '
350 W   T   W   F   W   3   T   W    @!'
351 W   T   W   F   W   3   T   W    @ rrv+2,cct say 'Please enter the';
351 W   T   W   F   W   3   T   W    + ' village name.'
352 W   T   W   F   W   3   T   W    read
353 W   T   W   F   W   3   T   W    IF lastkey()=27
354 <-------------------------------------------------return
355 W   T   W   F   W   3   T   W    ENDIF
356 W   T   W   F   W   3   T    ENDDO
357 W   T   W   F   W   3    ELSE
358 W   T   W   F   W   3   F    go top
359 W   T   W   F   ^=============loop
360 W   T   W   F   W   3.   ENDIF
361 W   T   W   F   W    CASE at(subst(nplace2,1,1),;
361 W   T   W   F   W    'ABCDEFGHIJKLMNOPQRSTUVWXYZ')>0
362 W   T   W   F   W   4    letter=subst(nplace2,1,1)
363 W   T   W   F   W   4    seek nprov+ndist+letter
364 W   T   W   F   W   4    IF .not. found()         && find the nearest;
364 W   T   W   F   W   4     findable letter in list
365 W   T   W   F   W   4   T    DO WHILE asc(letter)>=64.and..not.found()
366 W   T   W   F   W   4   T   W    letter=chr(asc(letter)-1)
367 W   T   W   F   W   4   T   W    seek nprov+ndist+letter
368 W   T   W   F   W   4   T    ENDDO
369 W   T   W   F   W   4    ENDIF
370 W   T   W   F   W   4    IF .not. found()         && if entered, symbols;
370 W   T   W   F   W   4     still won't be found
371 W   T   W   F   W   4   T    go top  && start over
372 W   T   W   F   W   4    ENDIF
373 W   T   W   F   ^=========loop
374 W   T   W   F   W    CASE (subst(nplace2,1,1)<>'0'.and.val(subst(nplace2,;
374 W   T   W   F   W    1,1))=0) .and. subst(nplace2,1,1)<>'*'   && bad entry
375 W   T   W   F   W    OTHERWISE        && number or asterisk entered
376 W   T   W   F   W   0    nmberplace=val(nplace2)
377 W   T   W   F   W   0    nplace=subst(str(1000+nmberplace,4),2)
378 W   T   W   F   W    ENDCASE
379 W   T   W   F<========EXIT
380 W   T   W   F    ENDDO choose nplace
381 W   T   W   F    set order to 2
382 W   T   W   F    seek nprov+ndist+nplace
383 W   T   W   F    IF .not. found()
384 W   T   W   F   T    ? chr(7)
385 W   T   W   F   T    nplace=space(3)
386 W   T   W   F   T    set order to 1
387 W   T   W   F   T    seek nprov+ndist
388 W   T   ^=============loop
389 W   T   W   F    ELSE
390 W   T   W   F   F    set order to 1
391 W   T   W   F   F    IF nplace<>'***'
392 W   T   W   F   F   T    cplace=place
393 W   T   W   F   F    ENDIF
394 W   T   W   F    ENDIF
395 W   T   W    ENDIF
396 W   T    ENDDO
397 W    ENDIF   for Vchoose=.T.
398 W    IF answer=space(1)
399 <========EXIT    && from choice of location
400 W    ENDIF
401 ENDDO
402
```

++ End of Program LOCATION ++ DOC found no errors.

VARIABLES: LOCAL CONCORDANCE FOR LOCATION

Variable	Statement number in listing (modified variables indicated by '=')										
answer:	=65	=67	72	75	=152	=155	160	163	166	=256	=263
	268	270	282	=339	=341	346	398				
ccs:	37	44	55	58	59	61	62	85	86	91	109
	143	146	147	149	173	174	175	184	185	192	193
	211	254	257	258	259	277	278	279	287	288	295
	302	303	321	335	338	348	350				
cct:	=37	66	67	92	93	94	95	96	97	98	99
	100	101	102	103	104	105	106	107	153	154	155
	196	201	206	207	260	261	262	263	306	311	316
	317	340	341	351							
cdist:	61	=83	=130	147	=171	=182	=240	=243	258	=275	=285
cplace:	62	=84	=132	149	=172	=183	=244	259	=276	=286	=294
	=337	=347	349	=350	=392						
cprov:	59	=82	=122	=125	=127	146	=170	257	=274		
Dchoose:	=42	=47	=73	=76	138	=177	=280	=290			
DISTNO:	196	305	310	329							
DISTRICT:	196	240									
letter:	=362	363	365	=366	367						
longlist:	=190	=203	205	=300	=313	315					
ndist:	46	61	=80	85	=129	144	147	148	160	=168	=180
	184	=225	231	=234	241	252	258	=272	=283	296	305
	310	329	330	363	367	382	387				
nmberplace:	=376	377									
nplace:	46	62	=81	86	=131	149	=169	=181	185	=242	255
	259	=273	=284	=293	295	=336	=377	382	=385	391	
nplace2:	=319	=321	334	361	362	374	376				
nprov:	46	56	58	60	72	=79	=111	116	=118	123	124
	141	146	=167	186	195	200	219	220	231	236	252
	257	=271	296	305	310	329	330	363	367	382	387
numberdist:	=209	=211	225								
numberprov:	=108	=109	111								
Pchoose:	=42	=47	51	=176	=280	=289					
PLACE:	306	392									
PLACENO:	306										
PROVINCE:	122										
PROVNO:	195	200	219	305	310	329					
rr:	=57	=63	66	67	=145	=150	=194	195	196	=197	201
	=304	305	306	=307	311						
rrd:	=39	61	85	147	174	184	193	211	258	278	287
rrp:	=38	57	58	59	91	109	146	173	257	277	
rrv:	=40	62	63	86	149	153	154	155	175	185	259
	260	261	262	263	279	288	295	303	321	338	340
	341	350	351								
Vchoose:	=42	=48	=73	=76	=161	=164	=177	250	=280	=290	

```
1 * sector.prg - identify sector of a project
2 * ver 1 Ron Barney 23 October 1990
3 * General logic:  ala, alb, alc and ald--sector category levels--are to be
4 * chosen in order.  Subroutines for making these choices are controlled
5 * within the main loop lachoose, lbchoose, lcchoose, ldchoose.  Exit from
6 * any of the subroutines can determine new values for these four variables.
7 * If remaining variable(s) .F., control goes back to the top of the loop,
8 * the current defaults display, and appropriate subroutines are executed.
9 * Exit only if ald choice accepted.
10
11 * Clipper function lastkey() has been used to test for the ESC key
12 * (ASCII 27) both here and in calling program ENTRY.  IF ESC is read from
13 * the keyboard buffer, the RETURN command is always issued.  While it is
14 * normally better structured programming to handle this kind of thing
15 * differently, there are are so many reads involved that this exceptional
16 * handling seems to make the code more clear, rather than the reverse.
17
18 ****************************
19 * select 5
20 * use ala index ala
21 * select 6
22 * use alb index alb      && key ala+alb
23 * select 7
24 * use alc index alc && key ala+alb+alc
25 * select 8
26 * use ald index aldname, ald     && keys ala+alb+alc+aldname, ala+alb+alc+
26 * ald
27
28 * default variables for sector
29 * store space(1) to nala,nalb,nalc                          && code 'number'
29 *  part
30 * nald=space(2)
31 * store space(31) to cala,calb,calc,cald          && code character
31 *  description
32 ********************************************
33
34 * ccs=20          && column where < sign will appear.  Assigned in calling
34 *  program
35 cct=ccs+2         && column where location text will begin
36 rrala=6 && row for ala
37 rralb=7 && row for alb
38 rralc=8 && row for alc
39 rrald=9 && row for ald
40
41 store .T. to lachoose, lbchoose, lcchoose, ldchoose
42
43 DO WHILE .T.
44 W     @ 0,ccs CLEAR TO 24,79
45 W     * DISPLAY DEFAULT SECTOR FOR CONFIRMATION
46 W     IF;
46 W      nala<>space(1).and.nalb<>space(1).and.nalc<>space(1).and.nald<>space;
46 W     (2)
47 W    T     store .F. to lachoose, lbchoose, lcchoose
48 W    T     ldchoose=.T.
49 W     ENDIF
50 W
51 W     IF lachoose      && LEVEL ONE;
51 W      ****************************************
52 W    T     select ala       && no need to go top.  Display is from program
53 W    T     DO WHILE .T.
54 W    T    W     @ 0,ccs clear to 24,79
```

```
55 W   T   W    IF nala<>space(1)        && default value has been set up
56 W   T   W   T    rr=rrala          && row for display of confirmation message
57 W   T   W   T    @ rrala,ccs-2 say nala
58 W   T   W   T    @ rrala,ccs say '< '+cala
59 W   T   W   T    IF nala='O'
60 W   T   W   T   T    @ rralb,ccs-2 say nalb+' < '+calb
61 W   T   W   T   T    @ rralc,ccs-2 say nalc+' < '+calc
62 W   T   W   T   T    @ rrald,ccs-3 say nald+' < '+cald
63 W   T   W   T   T    rr=rrald
64 W   T   W   T    ENDIF
65 W   T   W   T    answer=' '
66 W   T   W   T    @ rr+2,cct say 'Enter blank to confirm,'
67 W   T   W   T    @ rr+3,cct say 'any other key to reselect -->' get answer
68 W   T   W   T    read
69 W   T   W   T    DO CASE
70 W   T   W   T    CASE lastkey()=27
71 <-----------------------return
72 W   T   W   T    CASE answer=' '.and.nala='O'
73 W   T   W   T   2    store .F. to lbchoose, lcchoose, ldchoose
74 W   T<===============EXIT
75 W   T   W   T    CASE answer=' '
76 W   T   W   T   3    store .T. to lbchoose, lcchoose, ldchoose
77 W   T<===============EXIT
78 W   T   W   T    OTHERWISE
79 W   T   W   T   O    store space(1) to nala,nalb,nalc
80 W   T   W   T   O    store space(2) to nald
81 W   T   W   T   O    store space(31) to cala,calb,calc,cald
82 W   T   W   T   O    @ rralb,ccs-2 get nalb
83 W   T   W   T   O    @ rralc,ccs-2 get nalc
84 W   T   W   T   O    @ rrald,ccs-3 get nald
85 W   T   W   T   O    clear gets
86 W   T   W   T   O    go top
87 W   T   ^============loop
88 W   T   W   T    ENDCASE
89 W   T   W    ELSE    && no default value in force
90 W   T   W   F    @ rrala,ccs say '<'
91 W   T   W   F    @  rrala,cct say '1 Relief'
92 W   T   W   F    @  rrala+1,cct say '2 Safety'
93 W   T   W   F    @  rrala+2,cct say '3 Infrastructure'
94 W   T   W   F    @  rrala+3,cct say '4 Production'
95 W   T   W   F    @  rrala+4,cct say '5 Health'
96 W   T   W   F    @  rrala+5,cct say '6 Education'
97 W   T   W   F    @  rrala+6,cct say '7 Public Policy and Administration'
98 W   T   W   F    @  rrala+7,cct say '8 Expression of Ideas/Arts/Pure';
98 W   T   W   F    + ' Sciences'
99 W   T   W   F    @  rrala,ccs-2 get nala pict '9'
100 W  T   W   F    read
101 W  T   W    ENDIF
102 W  T   W    IF lastkey()=27
103 <-------------------return
104 W  T   W    ENDIF
105 W  T   W    seek nala
106 W  T   W    IF .not. found()
107 W  T   W   T    nala=space(1)
108 W  T   W   T    ? chr(7)
109 W  T   ^=========loop
110 W  T   W    ELSE
111 W  T   W   F    cala=alaname
112 W  T   W    ENDIF
113 W  T    ENDDO
114 W    ENDIF for lachoose=.T.
```

```
115 W
116 W     IF lbchoose                 && LEVEL TWO;
116 W      ****************************************
117 W    T     select alb
118 W    T     seek nala
119 W    T     DO WHILE .T.
120 W    T    W     @ 0,ccs clear to 24,79
121 W    T    W     IF nalb<>space(1) && confirm choice
122 W    T    W    T     rr=rralb          && row for display of confirmation message
123 W    T    W    T     @ rrala,ccs-2 say nala+' < '+cala
124 W    T    W    T     @ rralb,ccs-2 say nalb+' < '+calb
125 W    T    W    T     IF nalb='0'
126 W    T    W    T    T     @ rralc,ccs-2 say nalc+' < '+calc
127 W    T    W    T    T     rr=rralc
128 W    T    W    T     ENDIF
129 W    T    W    T     answer=' '
130 W    T    W    T     @ rr+2,cct say 'Enter blank to confirm,'
131 W    T    W    T     @ rr+3,cct say 'A to reselect level A,'
132 W    T    W    T     @ rr+4,cct say 'B to reselect level B -->' get answer
132 W    T    W    T      pict '!'
133 W    T    W    T     read
134 W    T    W    T     DO CASE
135 W    T    W    T     CASE lastkey()=27
136 <-----------------------return
137 W    T    W    T     CASE answer=' '.and. nalb='0'    && answer OK, but unspec
138 W    T    W    T    2     store .F. to lcchoose,ldchoose
139 W    T<===============EXIT && choice of level 2
140 W    T    W    T     CASE answer=' ' && answer OK:  go to next level
141 W    T    W    T    3     store .T. to lcchoose,ldchoose
142 W    T<===============EXIT && choice of level 2
143 W    T    W    T     CASE answer='A' && go back to level A
144 W    T    W    T    4     store space(1) to nala,nalb
145 W    T    W    T    4     store space(31) to cala,calb
146 W    T    W    T    4     @ rrala,ccs-2 get nala
147 W    T    W    T    4     @ rralb,ccs-2 get nalb
148 W    T    W    T    4     clear gets
149 W    T    W    T    4     store .T. to lachoose
150 W    T    W    T    4     store .F. to lbchoose,lcchoose,ldchoose
151 W    T<===============EXIT && current loop to main DO WHILE
152 W    T    W    T     OTHERWISE         && repeat level B
153 W    T    W    T    0     store space(1) to nalb
154 W    T    W    T    0     store space(31) to calb
155 W    T    W    T    0     @ rralb,ccs-2 get nalb
156 W    T    W    T    0     seek nala         && remember, indexed on nala+nalb
157 W    T    ^============loop
158 W    T    W    T     ENDCASE
159 W    T    W     ELSE     && display values; make choice
160 W    T    W    F     @ 0,ccs clear to 24,79
161 W    T    W    F     @ rralb,ccs say '<'
162 W    T    W    F     rr=rralb          && row counter
163 W    T    W    F     DO WHILE rr<22 .and.ala=nala .and..not.eof()
164 W    T    W    F    W     @ rr,cct say alb+space(2)+albname
165 W    T    W    F    W     rr=rr+1
166 W    T    W    F    W     skip
167 W    T    W    F     ENDDO
168 W    T    W    F     @ rralb,ccs-2 GET nalb pict '9
169 W    T    W    F     read
170 W    T    W    F     IF lastkey()=27
171 <-----------------------return
172 W    T    W    F     ENDIF
173 W    T    W     ENDIF    && choose nalb
```

```
174 W    T    W     seek nala+nalb
175 W    T    W     IF .not.found()
176 W    T    W    T     ? chr(7)
177 W    T    W    T     nalb=space(1)
178 W    T    W    T     seek nala
179 W    T    ^=========loop
180 W    T    W     ELSE
181 W    T    W    F     calb=albname
182 W    T    W     ENDIF
183 W    T     ENDDO
184 W     ENDIF for lbchoose=.T.
185 W
186 W     IF lcchoose      && LEVEL THREE;
186 W      ****************************************
187 W    T     select alc
188 W    T     seek nala+nalb
189 W    T     DO WHILE .T.
190 W    T    W     @ 0,ccs clear to 24,79
191 W    T    W     IF nalc<>space(1)        && determine whether chosen coding;
191 W    T    W      choice OK
192 W    T    W    T     answer=' '
193 W    T    W    T     @ rrala,ccs-2 say nala+' < '+cala
194 W    T    W    T     @ rralb,ccs-2 say nalb+' < '+calb
195 W    T    W    T     @ rralc,ccs-2 say nalc+' < '+calc
196 W    T    W    T     @ rralc+2,cct say 'Enter blank to confirm,'
197 W    T    W    T     @ rralc+3,cct say  A to reselect level A,'
198 W    T    W    T     @ rralc+4,cct say 'B to reselect level B,'
199 W    T    W    T     @ rralc+5,cct say 'C to reselect level C -->' get answer;
199 W    T    W    T      pict '!'
200 W    T    W    T     read
201 W    T    W    T     DO CASE
202 W    T    W    T     CASE lastkey()=27
203 <-------------------------return
204 W    T    W    T     CASE answer=' '
205 W    T    W    T    2     ldchoose=.T.
206 W    T<==============EXIT && choice of level C
207 W    T    W    T     CASE answer='A' && return to level A
208 W    T    W    T    3     store space(1) to nala,nalb,nalc
209 W    T    W    T    3     store space(31) to cala,calb,calc
210 W    T    W    T    3     @ rrala,ccs-2 get nala
211 W    T    W    T    3     @ rralb,ccs-2 get nalb
212 W    T    W    T    3     @ rralc,ccs-2 get nalc
213 W    T    W    T    3     clear gets
214 W    T    W    T    3     store .T. to lachoose
215 W    T    W    T    3     store .F. to lbchoose,lcchoose,ldchoose
216 W    T<==============EXIT && current loop to main DO WHILE
217 W    T    W    T     CASE answer='B' && return to level B
218 W    T    W    T    4     store space(1) to nalb,nalc
219 W    T    W    T    4     store space(31) to calb,calc
220 W    T    W    T    4     @ rralb,ccs-2 get nalb
221 W    T    W    T    4     @ rralc,ccs-2 get nalc
222 W    T    W    T    4     clear gets
223 W    T    W    T    4     store .T. to lbchoose
224 W    T    W    T    4     store .F. to lachoose,lcchoose,ldchoose
225 W    T<==============EXIT && current loop to main DO WHILE
226 W    T    W    T     OTHERWISE                          && repeat level C
227 W    T    W    T    0     IF answer<>'C'
228 W    T    W    T    0    T     ? chr(7)
229 W    T    W    T    0     ENDIF
230 W    T    W    T    0     nalc=space(1)
231 W    T    W    T    0     calc=space(31)
```

```
232 W   T   W   T   O    @ rralc,ccs-2 get nalc
233 W   T   W   T   O    seek nala+nalb
234 W   T   ^============LOOP
235 W   T   W   T    ENDCASE
236 W   T   W    ELSE    && nalc=space(1):  display coding choices; get one
237 W   T   W   F    @ 0,ccs clear to 24,79
238 W   T   W   F    @ rralc,ccs say '<'
239 W   T   W   F    IF found()       && sought nala+nalb at beginning of loop
240 W   T   W   F   T    rr=rralc          && row counter
241 W   T   W   F   T    DO WHILE rr<21 .and.ala+alb=nala+nalb .and..not.eof()
242 W   T   W   F   T   W    @ rr,cct say alc+space(1)+alcname
243 W   T   W   F   T   W    rr=rr+1
244 W   T   W   F   T   W    skip
245 W   T   W   F   T    ENDDO
246 W   T   W   F   T    @ rralc,ccs-2 get nalc pict '9'
247 W   T   W   F   T    read
248 W   T   W   F   T    IF lastkey()=27
249 <-------------------------return
250 W   T   W   F   T    ENDIF
251 W   T   W   F   T    seek nala+nalb+nalc
252 W   T   W   F   T    IF .not. found()
253 W   T   W   F   T   T    ? chr(7)
254 W   T   W   F   T   T    nalc=space(1)
255 W   T   W   F   T   T    seek nala+nalb
256 W   T   ^================loop
257 W   T   W   F   T    ELSE
258 W   T   W   F   T   F    calc=alcname
259 W   T   W   F   T    ENDIF
260 W   T   W   F    ELSE    && nala+nalb not found() at beginning of loop
261 W   T   W   F   F    @ 0,ccs clear to 24,79
262 W   T   W   F   F    @ rrala,ccs say '< '+cala
263 W   T   W   F   F    @ rralb,ccs say '< '+calb
264 W   T   W   F   F    @ rralc,ccs say   '< This level has not yet been';
264 W   T   W   F   F    + ' specified.'
265 W   T   W   F   F    @ rralc+1,ccs say '  Press any key to reselect';
265 W   T   W   F   F    + ' level A...'
266 W   T   W   F   F    @ 23,0
267 W   T   W   F   F    set color to n/n
268 W   T   W   F   F    wait space(79)
269 W   T   W   F   F    set color to w/n
270 W   T   W   F   F    store space(1) to nala,nalb,nalc
271 W   T   W   F   F    store space(31) to cala,calb,calc
272 W   T   W   F   F    @ rrala,ccs-2 get nala
273 W   T   W   F   F    @ rralb,ccs-2 get nalb
274 W   T   W   F   F    @ rralc,ccs-2 get nalc
275 W   T   W   F   F    clear gets
276 W   T   W   F   F    store .T. to lachoose
277 W   T   W   F   F    store .F. to lbchoose,lcchoose,ldchoose
278 W   T   W   F   F    answer='X'       && avoid quitting the main loop...
279 W   T<===============EXIT && current loop to main DO WHILE
280 W   T   W   F    ENDIF   && nala+nalb found()
281 W   T   W    ENDIF   && nalc=space(1)
282 W   T    ENDDO   while .T.
283 W    ENDIF   for lcchoose=.T.
284 W
285 W    IF ldchoose      && LEVEL FOUR;
285 W     ****************************************
286 W   T    select ald
287 W   T    set order to 1
288 W   T    IF nald=space(2)
289 W   T   T    seek nala+nalb+nalc
```

```
290 W    T     ELSE
291 W    T    F     seek nala+nalb+nalc+cald
292 W    T     ENDIF
293 W    T     DO WHILE .T.
294 W    T    W     @ 0,ccs clear to 24,79
295 W    T    W     IF nald<>space(2)
296 W    T    W    T     answer=' '
297 W    T    W    T     @ rrala,ccs-2 say nala+' < '+cala
298 W    T    W    T     @ rralb,ccs-2 say nalb+' < '+calb
299 W    T    W    T     @ rralc,ccs-2 say nalc+' < '+calc
300 W    T    W    T     @ rrald,ccs-3 say nald+' < '+cald
301 W    T    W    T     @ rrald+2,cct say 'Enter blank to confirm,'
302 W    T    W    T     @ rrald+3,cct say 'A to reselect level A,'
303 W    T    W    T     @ rrald+4,cct say 'B to reselect level B,'
304 W    T    W    T     @ rrald+5,cct say 'C to reselect level C,'
305 W    T    W    T     @ rrald+6,cct say 'D to reselect level D -->' get answer;
305 W    T    W    T      pict '!'
306 W    T    W    T     read
307 W    T    W    T     DO CASE
308 W    T    W    T     CASE lastkey()=27
309 <----------------------return
310 W    T    W    T     CASE answer=' '
311 W    T<===============EXIT && choice of level four
312 W    T    W    T     CASE answer='A' && return to level A
313 W    T    W    T    3     store space(1) to nala,nalb,nalc
314 W    T    W    T    3     nald=space(2)
315 W    T    W    T    3     store space(31) to cala,calb,calc,cald
316 W    T    W    T    3     @ rrala,ccs-2 get nala
317 W    T    W    T    3     @ rralb,ccs-2 get nalb
318 W    T    W    T    3     @ rralc,ccs-2 get nalc
319 W    T    W    T    3     @ rrald,ccs-3 get nald
320 W    T    W    T    3     clear gets
321 W    T    W    T    3     store .T. to lachoose
322 W    T    W    T    3     store .F. to lbchoose,lcchoose,ldchoose
323 W    T<===============EXIT && current loop to main DO WHILE
324 W    T    W    T     CASE answer='B' && return to level B
325 W    T    W    T    4     store space(1) to nalb,nalc
326 W    T    W    T    4     nald=space(2)
327 W    T    W    T    4     store space(31) to calb,calc,cald
328 W    T    W    T    4     @ rralb,ccs-2 get nalb
329 W    T    W    T    4     @ rralc,ccs-2 get nalc
330 W    T    W    T    4     @ rrald,ccs-3 get nald
331 W    T    W    T    4     clear gets
332 W    T    W    T    4     store .T. to lbchoose
333 W    T    W    T    4     store .F. to lachoose,lcchoose,ldchoose
334 W    T<===============EXIT && current loop to main DO WHILE
335 W    T    W    T     CASE answer='C' && return to level C
336 W    T    W    T    5     store space(1) to nalc
337 W    T    W    T    5     nald=space(2)
338 W    T    W    T    5     store space(31) to calc,cald
339 W    T    W    T    5     @ rralc,ccs-2 get nalc
340 W    T    W    T    5     @ rrald,ccs-3 get nald
341 W    T    W    T    5     clear gets
342 W    T    W    T    5     store .T. to lcchoose
343 W    T    W    T    5     store .F. to lachoose,lbchoose,ldchoose
344 W    T<===============EXIT && current loop to main DO WHILE
345 W    T    W    T     OTHERWISE                          && repeat level D
346 W    T    W    T    0     IF answer<>'D'
347 W    T    W    T    0    T     ? chr(7)
348 W    T    W    T    0     ENDIF
349 W    T    W    T    0     nald=space(2)
```

```
350 W   T   W   T   O    cald=space(31)
351 W   T   W   T   O    @ rrald,ccs-3 get nald
352 W   T   W   T   O    clear gets
353 W   T   W   T   O    set order to 1
354 W   T   W   T   O    seek nala+nalb+nalc
355 W   T   ^=============LOOP
356 W   T   W   T     ENDCASE
357 W   T   W    ELSE     && nald=space(2): display coding choices; get one
358 W   T   W   F     IF found()       && sought nala+nalb+nalc, beginning of;
358 W   T   W   F      loop/reselect D
359 W   T   W   F   T    longlist=.F.     && controls display of scroll message
360 W   T   W   F   T    DO WHILE .T.
361 W   T   W   F   T   W    @ 0,ccs clear to 24,79
362 W   T   W   F   T   W    @ rrald,ccs say '<'
363 W   T   W   F   T   W    rr=1     && row counter
364 W   T   W   F   T   W    DO WHILE rr<21 .and.ala+alb+alc=nala+nalb+nalc;
364 W   T   W   F   T   W      .and..not.eof()
365 W   T   W   F   T   W   W    @ rr,cct say ald+space(1)+aldname
366 W   T   W   F   T   W   W    rr=rr+1
367 W   T   W   F   T   W   W    skip
368 W   T   W   F   T   W    ENDDO
369 W   T   W   F   T   W    IF ala+alb+alc<>nala+nalb+nalc.or.eof()
370 W   T   W   F   T   W   T    @ rr,cct+3 say '*** End of list ***'
371 W   T   W   F   T   W    ELSE
372 W   T   W   F   T   W   F    longlist=.T.
373 W   T   W   F   T   W    ENDIF
374 W   T   W   F   T   W    IF longlist
375 W   T   W   F   T   W   T    @ 22,cct say 'Enter number to select,';
375 W   T   W   F   T   W   T    + ' letter for'
376 W   T   W   F   T   W   T    @ 23,cct say 'lookup, '+chr(25)+' to see;
376 W   T   W   F   T   W   T     more codes.'
377 W   T   W   F   T   W    ENDIF
378 W   T   W   F   T   W    nald2=space(2)  && temporary variable, to be;
378 W   T   W   F   T   W     changed to value
379 W   T   W   F   T   W    readexit(.T.)   && clipper command
380 W   T   W   F   T   W    @ rrald,ccs-3 get nald2 pict '@! X9'
381 W   T   W   F   T   W    read
382 W   T   W   F   T   W    readexit(.F.)   && clipper command
383 W   T   W   F   T   W    DO CASE
384 W   T   W   F   T   W    CASE lastkey()=27
385 <-------------------------------return;
386 W   T   W   F   T   W   1    *
386 W   T   W   F   T   W   1    *              CASE readkey()=5 .or.
386 W   T   W   F   T   W   1    *  readkey()=261     && down arrow (dbase)
387 W   T   W   F   T   W    CASE lastkey()=24       && down arrow entered;
387 W   T   W   F   T   W     (clipper)
388 W   T   W   F   T   W   2    IF (ala+alb+alc<>nala+nalb+nalc).or.eof()
389 W   T   W   F   T   W   2   T    seek nala+nalb+nalc     && always found
390 W   T   W   F   T   W   2   T    * else continue from where you are.
391 W   T   W   F   T   W   2    ENDIF
392 W   T   W   F   T   ^=========loop
393 W   T   W   F   T   W    CASE at(subst(nald2,1,1),;
393 W   T   W   F   T   W    'ABCDEFGHIJKLMNOPQRSTUVWXYZ')>0
394 W   T   W   F   T   W   3    letter=subst(nald2,1,1)
395 W   T   W   F   T   W   3    seek nala+nalb+nalc+letter
396 W   T   W   F   T   W   3    IF .not. found()         && find the nearest;
396 W   T   W   F   T   W   3     findable letter in list
397 W   T   W   F   T   W   3   T    DO WHILE asc(letter)>=64.and..not.found()
398 W   T   W   F   T   W   3   T   W    letter=chr(asc(letter)-1)
399 W   T   W   F   T   W   3   T   W    seek nala+nalb+nalc+letter
400 W   T   W   F   T   W   3   T    ENDDO
```

```
401 W   T   W   F   T   W   3    ENDIF
402 W   T   W   F   T   W   3    IF .not. found()          && if entered,;
402 W   T   W   F   T   W   3     symbols still won't be found
403 W   T   W   F   T   W   3   T    go top  && start over
404 W   T   W   F   T   W   3    ENDIF
405 W   T   W   F   T   ^=========loop
406 W   T   W   F   T   W    CASE (subst(nald2,1,1)<>'0'.and.val(subst(nald2,;
406 W   T   W   F   T   W    1,1))=0)          && bad entry
407 W   T   W   F   T   W    OTHERWISE         && number or asterisk entered
408 W   T   W   F   T   W   0    nald2=val(nald2)
409 W   T   W   F   T   W   0    nald=subst(str(100+nald2,3),2)
410 W   T   W   F   T   W    ENDCASE
411 W   T   W   F   T<========EXIT
412 W   T   W   F   T    ENDDO choose nald
413 W   T   W   F   T    set order to 2
414 W   T   W   F   T    seek nala+nalb+nalc+nald
415 W   T   W   F   T    IF .not. found()
416 W   T   W   F   T   T    ? chr(7)
417 W   T   W   F   T   T    nald=space(2)
418 W   T   W   F   T   T    set order to 1
419 W   T   W   F   T   T    seek nala+nalb+nalc
420 W   T   ^================loop
421 W   T   W   F   T    ELSE
422 W   T   W   F   T   F    cald=aldname
423 W   T   W   F   T   F    set order to 1
424 W   T   W   F   T    ENDIF
425 W   T   W   F    ELSE    && nala+nalb+nalc .not. found()
426 W   T   W   F   F    @ 0,ccs clear to 24,79
427 W   T   W   F   F    @ rrala,ccs say '< '+cala
428 W   T   W   F   F    @ rralb,ccs say '< '+calb
429 W   T   W   F   F    @ rralc,ccs say '< '+calc
430 W   T   W   F   F    @ rrald,ccs say  '< This level has not yet been';
430 W   T   W   F   F    + ' specified.'
431 W   T   W   F   F    @ rrald+1,ccs say '  Press any key to reselect';
431 W   T   W   F   F    + ' level A...'
432 W   T   W   F   F    @ 23,0
433 W   T   W   F   F    set color to n/n
434 W   T   W   F   F    wait space(79)
435 W   T   W   F   F    set color to w/n
436 W   T   W   F   F    store space(1) to nala,nalb,nalc                  &&;
436 W   T   W   F   F     nald already blank
437 W   T   W   F   F    store space(31) to cala,calb,calc        && cald;
437 W   T   W   F   F     already blank
438 W   T   W   F   F    @ rrala,ccs-2 get nala
439 W   T   W   F   F    @ rralb,ccs-2 get nalb
440 W   T   W   F   F    @ rralc,ccs-2 get nalc
441 W   T   W   F   F    clear gets
442 W   T   W   F   F    store .T. to lachoose
443 W   T   W   F   F    store .F. to lbchoose,lcchoose,ldchoose
444 W   T   W   F   F    answer='X'       && avoid quitting the main loop...
445 W   T<================EXIT && current loop to main DO WHILE
446 W   T   W   F    ENDIF for nala+nalb+nalc found()
447 W   T   W    ENDIF for nald<>space(2)
448 W   T    ENDDO
449 W    ENDIF   for ldchoose=.T.
450 W
451 W    IF answer=space(1)
452 <=========EXIT    && from choice of location
453 W    ENDIF
454 ENDDO
455
```

++ End of Program SECTOR ++ DOC found no errors.

VARIABLES: LOCAL CONCORDANCE FOR SECTOR

Variable	Statement number in listing (modified variables indicated by '=')										
ALA:	163	241	364	369	388						
ALANAME:	111										
ALB:	164	241	364	369	388						
ALBNAME:	164	181									
ALC:	242	364	369	388							
ALCNAME:	242	258									
ALD:	365										
ALDNAME:	365	422									
answer:	=65	=67	72	75	=129	=132	137	140	143	=192	=199
	204	207	217	227	=278	=296	=305	310	312	324	335
	346	=444	451								
cala:	58	=81	=111	123	=145	193	=209	262	=271	297	=315
	427	=437									
calb:	60	=81	124	=145	=154	=181	194	=209	=219	263	=271
	298	=315	=327	428	=437						
calc:	61	=81	126	195	=209	=219	=231	=258	=271	299	=315
	=327	=338	429	=437							
cald:	62	=81	291	300	=315	=327	=338	=350	=422		
ccs:	35	44	54	57	58	60	61	62	82	83	84
	90	99	120	123	124	126	146	147	155	160	161
	168	190	193	194	195	210	211	212	220	221	232
	237	238	246	261	262	263	264	265	272	273	274
	294	297	298	299	300	316	317	318	319	328	329
	330	339	340	351	361	362	380	426	427	428	429
	430	431	438	439	440						
cct:	=35	66	67	91	92	93	94	95	96	97	98
	130	131	132	164	196	197	198	199	242	301	302
	303	304	305	365	370	375	376				
lachoose:	=41	=47	51	=149	=214	=224	=276	=321	=333	=343	=442
lbchoose:	=41	=47	=73	=76	116	=150	=215	=223	=277	=322	=332
	=343	=443									
lcchoose:	=41	=47	=73	=76	=138	=141	=150	186	=215	=224	=277
	=322	=333	=342	=443							
ldchoose:	=41	=48	=73	=76	=138	=141	=150	=205	=215	=224	=277
	285	=322	=333	=343	=443						
letter:	=394	395	397	=398	399						
longlist:	=359	=372	374								
n:	267	269	433	435							
nala:	46	55	57	59	72	=79	=99	105	=107	118	123
	=144	=146	156	163	174	178	188	193	=208	=210	233
	241	251	255	=270	=272	289	291	297	=313	=316	354
	364	369	388	389	395	399	414	419	=436	=438	
nalb:	46	60	=79	=82	121	124	125	137	=144	=147	=153
	=155	=168	174	=177	188	194	=208	=211	=218	=220	233
	241	251	255	=270	=273	289	291	298	=313	=317	=325
	=328	354	364	369	388	389	395	399	414	419	=436
	=439										
nalc:	46	61	=79	=83	126	191	195	=208	=212	=218	=221
	=230	=232	=246	251	=254	=270	=274	289	291	299	=313
	=318	=325	=329	=336	=339	354	364	369	388	389	395
	399	414	419	=436	=440						
nald:	46	62	=80	=84	288	295	300	=314	=319	=326	=330
	=337	=340	=349	=351	=409	414	=417				
nald2:	=378	=380	393	394	406	=408	409				
rr:	=56	=63	66	67	=122	=127	130	131	132	=162	163

VARIABLES: LOCAL CONCORDANCE FOR SECTOR (continued)

Variable	Statement number in listing (modified variables indicated by '=')										
	164	=165	=240	241	242	=243	=363	364	365	=366	370
rrala:	=36	56	57	58	90	91	92	93	94	95	96
	97	98	99	123	146	193	210	262	272	297	316
	427	438									
rralb:	=37	60	82	122	124	147	155	161	162	168	194
	211	220	263	273	298	317	328	428	439		
rralc:	=38	61	83	126	127	195	196	197	198	199	212
	221	232	238	240	246	264	265	274	299	318	329
	339	429	440								
rrald:	=39	62	63	84	300	301	302	303	304	305	319
	330	340	351	362	380	430	431				
w:	269	435									

```
1 * cont.prg - choose contractor
2 * ver 1.0 Ron Barney 25 October 1990
3 * Clipper function lastkey() has been used to test for the ESC key
4 * (ASCII 27) both here and in calling program ENTRY.  IF ESC is read from
5 * the keyboard buffer, the RETURN command is always issued.  While it is
6 * normally better structured programming to handle this kind of thing
7 * differently, there are are so many reads involved that this exceptional
8 * handling seems to make the code more clear, rather than the reverse.
9
10 * Environment:
11 * select 9
12 * use contract index conname, conno
13 * select 10
14 * use repdata
15 * ncon=dfconno
16 * ccon=dfconname
17 * vconabbrev initialized as abbreviation of dfconname
18 * ccs=20          && column where < sign will appear.  Assigned in calling
18 *  program
19 * cct=ccs+2       && column where location text will begin
20
21 rrc=12            && row where contractor will appear
22
23 select contract
24 go top
25 DO WHILE .T.
26 W    @ 0,ccs clear to 24,79
27 W    IF ncon<>space(2)
28 W   T    answer=' '
29 W   T    @ rrc,ccs-3 say ncon+' < '+ccon
30 W   T    @ rrc+2,cct say 'Enter blank to confirm,'
31 W   T    @ rrc+3,cct say 'Any other key to reselect -->' get answer pict ';
31 W   T    !'
32 W   T    read
33 W   T    IF answer=' '.or.lastkey()=27
34 <-------------return
35 W   T    ELSE
36 W   T   F    ncon=space(2)
37 W   T   F    ccon=space(42)  && label for conname
38 W   T   F    @ rrc,ccs-3 say ncon
39 W   T   F    go top
40 ^============loop
41 W   T    ENDIF
42 W    ELSE    && ncon=space(2)
43 W   F    longlist=.F.    && controls display of scroll message
44 W   F    DO WHILE .T.
45 W   F   W    @ 0,ccs clear to 24,79
46 W   F   W    @ rrc,ccs say '<'
47 W   F   W    rr=1    && row counter
48 W   F   W    DO WHILE rr<21 .and..not.eof()
49 W   F   W   W    @ rr,cct say conno+space(1)+conname
50 W   F   W   W    rr=rr+1
51 W   F   W   W    skip
52 W   F   W    ENDDO
53 W   F   W    IF eof()
54 W   F   W   T    @ rr,cct say '*** End of list ***'
55 W   F   W    ELSE
56 W   F   W   F    longlist=.T.
57 W   F   W    ENDIF
58 W   F   W    IF longlist
59 W   F   W   T    @ 22,cct say 'Enter number to select, letter for name'
```

```
 60 W    F    W    T     @ 23,cct say 'lookup, '+chr(25)+' to see more names.'
 61 W    F    W     ENDIF
 62 W    F    W     ncon2=space(2)
 63 W    F    W     readexit(.T.)    && clipper command
 64 W    F    W     @ rrc,ccs-3 get ncon2 pict '@! XX'
 65 W    F    W     read
 66 W    F    W     readexit(.F.)    && clipper command
 67 W    F    W     DO CASE
 68 W    F    W     CASE lastkey()=27
 69 <-----------------return
 70 W    F    W    1     *                                CASE readkey()=5 .or.
 70 W    F    W    1     *  readkey()=261      && down arrow (dbase)
 71 W    F    W     CASE lastkey()=24        && down arrow entered (clipper)
 72 W    F    W    2     IF eof()
 73 W    F    W    2    T     go top
 74 W    F    W    2    T     * else continue from where you are.
 75 W    F    W    2     ENDIF
 76 W    F   ^========loop
 77 W    F    W     CASE at(subst(ncon2,1,1),'ABCDEFGHIJKLMNOPQRSTUVWXYZ')>0
 78 W    F    W    3     letter=subst(ncon2,1,1)
 79 W    F    W    3     seek letter
 80 W    F    W    3     IF .not. found()         && find the nearest findable;
 80 W    F    W    3      letter in list
 81 W    F    W    3    T     DO WHILE asc(letter)>=64.and..not.found()
 82 W    F    W    3    T    W     letter=chr(asc(letter)-1)
 83 W    F    W    3    T    W     seek letter
 84 W    F    W    3    T     ENDDO
 85 W    F    W    3     ENDIF
 86 W    F    W    3     IF .not. found()         && if entered, symbols still won;
 86 W    F    W    3     't be found
 87 W    F    W    3    T     go top  && start over
 88 W    F    W    3     ENDIF
 89 W    F   ^========loop
 90 W    F    W     CASE (subst(ncon2,1,1)<>'0'.and.val(subst(ncon2,1,1))=0) &&;
 90 W    F    W      bad entry
 91 W    F    W     OTHERWISE        && number entered
 92 W    F    W    0     nconname=val(ncon2)
 93 W    F    W    0     ncon=subst(str(100+nconname,3),2)
 94 W    F    W     ENDCASE
 95 W    F<=======EXIT
 96 W    F     ENDDO choose ncon
 97 W    F     set order to 2
 98 W    F     seek ncon
 99 W    F     IF .not. found()
100 W    F    T     ? chr(7)
101 W    F    T     ncon=space(2)
102 W    F    T     set order to 1
103 W    F    T     go top
104 ^============loop
105 W    F     ELSE
106 W    F    F     set order to 1
107 W    F    F     ccon=conname
108 W    F    F     vconabbrev=conabbrev
109 W    F     ENDIF
110 W     ENDIF
111 ENDDO
112
```

++ End of Program CONT ++ DOC found no errors.

VARIABLES: LOCAL CONCORDANCE FOR CONT

Variable	Statement number in listing (modified variables indicated by '=')						
answer:	=28	=31	33				
ccon:	29	=37	=107				
ccs:	26	29	38	45	46	64	
cct:	30	31	49	54	59	60	
CONABBREV:	108						
CONNAME:	49	107					
CONNO:	49						
letter:	=78	79	81	=82	83		
longlist:	=43	=56	58				
ncon:	27	29	=36	38	=93	98	=101
ncon2:	=62	=64	77	78	90	92	
nconname:	=92	93					
rr:	=47	48	49	=50	54		
rrc:	=21	29	30	31	38	46	64
vconabbrev:	=108						

```
 1 * chparty.prg - choose party for intermediary
 2 * ver 1.0 Ron Barney 14 November 1990
 3 * Clipper function lastkey() has been used to test for the ESC key
 4 * (ASCII 27) both here and in calling program ENTRY.  IF ESC is read from
 5 * the keyboard buffer, the RETURN command is always issued.  While it is
 6 * normally better structured programming to handle this kind of tning
 7 * differently, there are are so many reads involved that this exceptional
 8 * handling seems to make the code more clear, rather than the reverse.
 9  .
10 * ccs=22
11 * ccon=space(42)
12 * vpartyno=space(2)
13
14 select party    && index partv, partyno
15 set order to 1
16
17 @ 10,39 clear to 20,76
18 @ 10,39 to 20,76
19 DO WHILE .T.
20 W
21 W     *** DISPLAY PARTIES ***
22 W     @ 11,40 clear to 19,75
23 W     rr=11
24 W     DO WHILE rr<17
25 W   W     IF eof()
26 W   W   T     @ rr,41 say '** End of list **'
27 W   W   T     rr=rr+1
28 W   W   T     IF rr=17
29 W<==============EXIT
30 W   W   T     ENDIF
31 W   W   T     go top
32 W   W     ENDIF
33 W   W     @ rr,41 say partyno+' '+party
34 W   W     skip
35 W   W     rr=rr+1
36 W     ENDDO
37 W     @ 18,41 say 'Enter number to select, letter'
38 W     @ 19,41 say 'for lookup, '+chr(25)+' to scroll.'
39 W
40 W     readexit(.T.)   && clipper command
41 W     vpartyno=space(2)
42 W     @ 22,41 get vpartyno pict '@! X9'
43 W     read
44 W     readexit(.T.)   && clipper command
45 W
46 W     *** EVALUATE AND TAKE ACTION ***
47 W     DO CASE
48 W     CASE lastkey()=27
49 <---------return
50 W   1     *                CASE readkey()=5 .or. readkey()=261      && down
50 W   1     *  arrow (dbase)
51 W     CASE lastkey()=24        && down arrow entered (clipper)
52 ^========LOOP
53 W     CASE at(subst(vpartyno,1,1),'ABCDEFGHIJKLMNOPQRSTUVWXYZ')>0
54 W   3     letter=subst(vpartyno,1,1)
55 W   3     seek letter
56 W   3     IF .not. found()         && find the nearest findable letter in;
56 W   3      list
57 W   3   T     DO WHILE asc(letter)>=64.and..not.found()
58 W   3   T   W     letter=chr(asc(letter)-1)
59 W   3   T   W     seek letter
```

```
60 W   3   T    ENDDO
61 W   3    ENDIF
62 W   3    IF .not. found()          && if entered, symbols still won't be';
62 W   3    + ' found
63 W   3   T    go top  && start over
64 W   3    ENDIF
65 ^=========loop
66 W    CASE (subst(vpartyno,1,1)<>'0 .and.val(subst(vpartyno,1,1))=0) .and.;
66 W     subst(vpartyno,1,1)<>'*'  && bad entry
67 W    OTHERWISE        && number or asterisk entered
68 W   0    set order to 2
69 W   0    vpartyno=subst(str(val(vpartyno)+100,3),2)
70 W   0    seek vpartyno
71 W   0    IF found()
72 W   0   T    dummy=vpartyno+' < '+trim(party)+'?'
73 W   0   T    answer='Y'
74 W   0   T    @ 22,41 say dummy get answer pict 'Y'
75 W   0   T    read
76 W   0   T    IF lastkey()=27
77 <-------------------return
78 W   0   T    ENDIF
79 W   0   T    IF answer='Y'
80 W   0   T   T    @ 22,41 say vpartyno+' < '+trim(party)+space(4)
81 W   0   T   T    vparty=party
82 <=========.=======EXIT
83 W   0   T    ENDIF
84 W   0    ELSE
85 W   0   F    ? chr(7)
86 W   0   F    go top
87 W   0    ENDIF
88 W   0    vpartyno=space(2)
89 W   0    @ 22,41 get vpartyno
90 W   0    clear gets
91 W   0    @ 22,43 say space(40)
92 W   0    set order to 1
93 W    ENDCASE
94 ENDDO
95 @ 10,39 clear to 20,76
96 @ 12,ccs say '< '+ccon
97 IF 'B'$(ald->indic).or.'C'$(ald->indic)
98 T    @ 13, 56  say 'Donated?' get vdonated pict 'Y'
99 ENDIF
100 @ 14,30 to 14,76
101 @ 15, 35  SAY "Male Afghan salaried beneficiaries"
102 @ 16, 35  SAY "Female Afghan salaried beneficiaries"
103 @ 17, 35  SAY "Non-Afghan salaried beneficiaries"
104 @ 18, 35  SAY "Unsalaried beneficiaries"
105 @ 19, 35  SAY "Indirect beneficiaries"
106 IF ald->vwafghs
107 T    @ 15, 72  get  vmafsbene pict '999999'
108 T    @ 16, 72  get  vfafsbene pict '999999'
109 ENDIF
110 IF ald->vwnafghs
111 T    @ 17, 72  get  vnafsbene pict '999999'
112 ENDIF
113 IF ald->vwunsal
114 T    @ 18, 70  get  vunsbene  pict '99999999'
115 ENDIF
116 IF ald->vwindir
117 T    @ 19, 70  get  vindbene  pict '99999999'
118 ENDIF
```

```
119
120 clear gets
121
```

++ End of Program CHPARTY ++ DOC found no errors.

VARIABLES: LOCAL CONCORDANCE FOR CHPARTY

Variable	Statement number in listing (modified variables indicated by '=')										
answer:	=73	=74	79								
ccon:	96										
ccs:	96										
dummy:	=72	74									
INDIC:	97										
letter:	=54	55	57	=58	59						
PARTY:	33	72	80	81							
PARTYNO:	33										
rr:	=23	24	26	=27	28	33	=35				
vdonated:	=98										
vfafsbene:	=108										
vindbene:	=117										
vmafsbene:	=107										
vnafsbene:	=111										
vparty:	=81										
vpartyno:	=41	=42	53	54	66	=69	70	72	80	=88	=89
vunsbene:	=114										
VWAFGHS:	106										
VWINDIR:	116										
VWNAFGHS:	110										
VWUNSAL:	113										

```
  1 * rptobjhd.prg - header file for full report of indicators by objective
  2 * ver 1.0 Ron Barney 18 November 1990
  3
  4 @ 0,1 say 'USAID/AFGHANISTAN: PERFORMANCE INDICATORS FOR Q'+qtrset+ ' 19'+;
  4 yearset+' BY OBJECTIVE'
  5 @ 0,191 say 'Page '+ltrim(str(pp))
  6 @ 2,1 say 'OBJECTIVE: '+indic->objective
  7 @ 2,89 say 'Dollar    '+ '|-------------------------Beneficiaries-------------;
  7 -------------|'
  8 @ 3,1 SAY 'INDICATOR: '+indic->indicator
  9 @ 3,78 say 'Dollars    Trans- '
 10 @ 4,80 say 'USAID    porta-    |-----------Salaried---------------|  Afgh.'
 11 @ 5,60 say 'Units of           Contri-    tation   Afghan  Afghan    Non-'+;
 11  '      Total    % of      Unsal-                  |---Project Dates--|    '+ ';
 11 Monitoring'
 12 @ 6, 7 say 'Item                                Origin        Amount ';
 12 + ' Measurement'+ '         bution       Cost    Males    Fem.   Afgh. ;
 12  Salaried  Total'+ '       aried   Indirect      Start      End       ;
 12  Status/Date'
 13 rr=8
 14 pp=pp+1
 15
```

++ End of Program RPTOBJHD ++ DOC found no errors.

VARIABLES: LOCAL CONCORDANCE FOR RPTOBJHD

Variable	Statement number in listing (modified variables indicated by '=')
INDICATOR:	8
OBJECTIVE:	6
pp:	5 =14
qtrset:	4
rr:	=13
yearset:	4

```
 1 * rptlochd.prg - header file for full report of indictors by location
 2 * ver 1.0 Ron Barney 20 November 1990
 3
 4 @ 0,1 say 'USAID/AFGHANISTAN: PERFORMANCE INDICATORS FOR Q'+qtrset+ ' 19'+;
 4 yearset+' BY LOCATION'
 5 @ 0,191 say 'Page '+ltrim(str(pp))
 6 IF province->provno<>'99'
 7 T     @ 2,1 say upper(trim(province->province))+' PROVINCE, '+;
 7 T     upper(trim(district->district))+' DISTRICT'
 8 ELSE
 9 F     @ 2,1 say 'PAKISTAN'
10 ENDIF
11 *****
12 @ 2,88 say 'Dollar  |----------------------Beneficiaries----------------------|'
13 @ 3,78 say 'Dollars   Trans-'
14 @ 4,80 say 'USAID   porta-  |------------Salaried---------| Afgh.'+space(43)+;
14  '|---------Monitoring----------|'
15 @ 5,60 say 'Units of          Contri-   tation  Afghan Afghan   Non-'+ ' ;
15          % of    Unsal-    Indi-  |---Project Dates--|'+ '  |------Status---;
15 ---|'
16 @ 6,7 say 'Item                                Origin        Amount ';
16 + ' Measurement'+ '         bution      Cost   Males   Fem.  Afgh.  Total;
16  Total      aried'+ '      rect    Start      End        1    0   -1    2  ;
16   Date'
17 rr=8
18 pp=pp+1
19
```

++ End of Program RPTLOCHD ++ DOC found no errors.

VARIABLES: LOCAL CONCORDANCE FOR RPTLOCHD

Variable	Statement number in listing (modified variables indicated by '=')
DISTRICT:	7
pp:	5 =18
PROVINCE:	7
PROVNO:	6
qtrset:	4
rr:	=17
yearset:	4

```
 1 * rebuild.prg - rebuild indexes
 2 * ver 1.0 Ron Barney 14 November 1990
 3
 4 .lear
 5 set safety off
 6 @ 4,10 say 'Creating indexes...'
 7
 8 @ 6,10 say 'Indexes for province.dbf'
 9 use province
10 index on province to province
11 index on provno to provno
12
13
14 @ 7,10 say 'Indexes for district.dbf'
15 use district
16 index on provno+district to pdistrict
17 index on provno+distno to pdistno
18
19 @ 8,10 say 'Indexes for place.dbf'
20 use place
21 index on provno+distno+place to pdplace
22 index on provno+distno+placeno to pdplacen
23
24 @ 9,10 say 'Indexes for ala.dbf'
25 use ALA
26 index on ALA to ALA
27
28 @ 10,10 say 'Indexes for alb.dbf'
29 use ALB
30 index on ALA+ALB to ALB
31
32 @ 11,10 say 'Indexes for alc.dbf'
33 use ALC
34 index on ALA+ALB+ALC to ALC
35
36 @ 12,10 say 'Indexes for ald.dbf'
37 use ALD
38 index on ALA+ALB+ALC+ALDname to ALDname
39 index on ALA+ALB+ALC+ALD to ALD
40
41 @ 13,10 say 'Indexes for contract.dbf'
42 use contract
43 index on conname to conname
44 index on conno to conno
45
46 @ 14,10 say 'Indexes for party.dbf'
47 use party
48 index on party to party
49 index on partyno to partyno
50
51 @ 15,10 say 'Indexes for repdata.dbf'
52 use repdata
53 index on qtr+year+provno+distno+placeno+ala+alb+alc+ald+conno+subgrant+;
53 seq to repdata
54 index on provno+distno+ala+alb+alc+ald+conabbrev to rpt1
55 set safety on
56 ? chr(7)
57
58
```

++ End of Program REBUILD ++ DOC found no errors.

VARIABLES: LOCAL CONCORDANCE FOR REBUILD

Variable	Statement number in listing (modified variables indicated by '=')						
ALA:	26	30	34	38	39	53	54
ALB:	30	34	38	39	53	54	
ALC:	34	38	39	53	54		
ALD:	39	53	54				
ALDNAME:	38						
CONABBREV:	54						
CONNAME:	43						
CONNO:	44	53					
DISTNO:	17	21	22	53	54		
DISTRICT:	16						
off:	5						
on:	55						
PARTY:	48						
PARTYNO:	49						
PLACE:	21						
PLACENO:	22	53					
PROVINCE:	10						
PROVNO:	11	16	17	21	22	53	54
QTR:	53						
SEQ:	53						
SUBGRANT:	53						
YEAR:	53						

VARIABLES: GLOBAL CONCORDANCE 212 variables

Variable occurs in	Modules (modified variables indicated by '=')
AIDAMT	=ENTRY RPTLOC RPTOBJ
ALA	=ENTRY REBUILD RPTLOC RPTOBJ SECTOR
ALANAME	RPTLOC RPTOBJ SECTOR
ALB	=ENTRY REBUILD RPTLOC RPTOBJ SECTOR
ALBNAME	RPTLOC RPTOBJ SECTOR
ALC	=ENTRY REBUILD RPTLOC RPTOBJ SECTOR
ALCNAME	RPTLOC RPTOBJ SECTOR
ALD	=ENTRY REBUILD RPTLOC RPTOBJ SECTOR
ALDNAME	REBUILD RPTLOC RPTOBJ SECTOR
answer	=CHPARTY =CONT =DEFAULT =ENTRY =LOCATION =REP =RPTLOC RPTOBJ =SECTOR =UTIL
BEGIN	=ENTRY RPTLOC RPTOBJ
cala	=ENTRY =SECTOR
calb	=ENTRY =SECTOR
calc	=ENTRY =SECTOR
cald	=ENTRY =SECTOR
ccon	CHPARTY =CONT =ENTRY
ccs	CHPARTY CONT =ENTRY LOCATION SECTOR
cct	CONT =ENTRY =LOCATION =SECTOR
cdist	=ENTRY =LOCATION
CONABBREV	CONT DEFAULT =ENTRY REBUILD RPTLOC RPTOBJ
CONNAME	CONT DEFAULT REBUILD
CONNO	CONT =ENTRY REBUILD
cplace	=ENTRY =LOCATION
cprov	=ENTRY =LOCATION
Dchoose	=LOCATION
df*	DEFAULT ENTRY
dfconabbre	=DEFAULT
dfconname	=DEFAULT ENTRY
dfconno	=DEFAULT ENTRY
dfprice	=DEFAULT =ENTRY
dfqtr	=DEFAULT ENTRY
dfsubgrant	=DEFAULT ENTRY
dfyear	=DEFAULT ENTRY
DISTNO	=ENTRY LOCATION REBUILD RPTLOC RPTOBJ
DISTRICT	LOCATION REBUILD RPTLOC RPTLOCHD RPTOBJ
disttot	=RPTLOC =RPTOBJ
DONATED	=ENTRY RPTLOC RPTOBJ
DPI	=ENTRY RPTLOC RPTOBJ
dtaidamt	=RPTLOC =RPTOBJ
dtfafsbene	=RPTLOC =RPTOBJ
dtindbene	=RPTLOC =RPTOBJ
dtmafsbene	=RPTLOC =RPTOBJ
dtnafsbene	=RPTLOC =RPTOBJ
dtstata	=RPTLOC
dtstatb	=RPTLOC
dtstatc	=RPTLOC
dtstatd	=RPTLOC
dttranspor	=RPTLOC =RPTOBJ
dtunsbene	=RPTLOC =RPTOBJ
dummy	=CHPARTY =ENTRY
END	=ENTRY RPTLOC RPTOBJ
facs	=RPTLOC
FAFSBENE	=ENTRY RPTLOC RPTOBJ
firstpass	=ENTRY
INDBENE	=ENTRY RPTLOC RPTOBJ

VARIABLES: GLOBAL CONCORDANCE (continued)

ariable occurs in	Modules (modified variables indicated by '=')
NDIC	CHPARTY ENTRY RPTLOC RPTOBJ
NDICATOR	RPTLOC RPTOBJHD
ndictot	=RPTLOC
NDNO	RPTLOC RPTOBJ
NTERMED	=ENTRY
taidamt	=RPTLOC
tfafsbene	=RPTLOC
tindbene	=RPTLOC
tmafsbene	=RPTLOC
tnafsbene	=RPTLOC
tstata	=RPTLOC
tstatb	=RPTLOC
tstatc	=RPTLOC
tstatd	=RPTLOC
ttranspor	=RPTLOC
tunsbene	=RPTLOC
achoose	=SECTOR
bchoose	=SECTOR
cchoose	=SECTOR
dchoose	=SECTOR
etter	=CHPARTY =CONT =LOCATION =SECTOR
onglist	=CONT =LOCATION =SECTOR
MAFSBENE	=ENTRY RPTLOC RPTOBJ
nala	=ENTRY
nalb	=ENTRY
nalc	=ENTRY
nald	=ENTRY
ncon	=ENTRY
ndist	=ENTRY
MEAS	ENTRY RPTLOC RPTOBJ
nplace	=ENTRY
nprov	=ENTRY
nqtr	=ENTRY
nseq	=ENTRY
nsubgrant	=ENTRY
nyear	=ENTRY
n	ENTRY SECTOR
NAFSBENE	=ENTRY RPTLOC RPTOBJ
nala	=ENTRY =SECTOR
nalb	=ENTRY =SECTOR
nalc	=ENTRY =SECTOR
nald	=ENTRY =SECTOR
nald2	=SECTOR
ncon	=CONT =ENTRY
ncon2	=CONT
nconname	=CONT
ndist	=ENTRY =LOCATION
newalc	=RPTLOC =RPTOBJ
newdist	=RPTLOC =RPTOBJ
newind	=RPTOBJ
newindic	=RPTLOC
newprov	=RPTLOC =RPTOBJ
nmberplace	=LOCATION
nplace	=ENTRY =LOCATION
nplace2	=LOCATION
nprov	=ENTRY =LOCATION
numberdist	=LOCATION

VARIABLES: GLOBAL CONCORDANCE (continued)

Variable occurs in	Modules (modified variables indicated by '=')
numberprov	=LOCATION
OBJECTIVE	RPTLOC RPTOBJHD
off	DEFAULT ENTRY REBUILD REP
on	DEFAULT ENTRY REBUILD
packdb	=ENTRY
PARTY	CHPARTY ENTRY REBUILD
PARTYNO	CHPARTY =ENTRY REBUILD
pass1	=RPTLOC =RPTOBJ
password	=UTIL
Pchoose	=LOCATION
PLACE	LOCATION REBUILD
PLACENO	=ENTRY LOCATION REBUILD
pp	=RPTLOC =RPTLOCHD =RPTOBJ =RPTOBJHD
PRICE	=ENTRY RPTLOC RPTOBJ
PRICEUNIT	ENTRY RPTLOC RPTOBJ
print	RPTLOC RPTOBJ
process	=ENTRY
PROVINCE	LOCATION REBUILD RPTLOCHD RPTOBJ
PROVNO	=ENTRY LOCATION REBUILD RPTLOC RPTLOCHD RPTOBJ
provtot	=RPTLOC =RPTOBJ
ptaidamt	=RPTLOC =RPTOBJ
ptfafsbene	=RPTLOC =RPTOBJ
ptindbene	=RPTLOC =RPTOBJ
ptmafsbene	=RPTLOC =RPTOBJ
ptnafsbene	=RPTLOC =RPTOBJ
ptstata	=RPTLOC
ptstatb	=RPTLOC
ptstatc	=RPTLOC
ptstatd	=RPTLOC
pttranspor	=RPTLOC =RPTOBJ
ptunsbene	=RPTLOC =RPTOBJ
QTR	=ENTRY REBUILD RPTLOC RPTOBJ
qtrset	=RPTLOC RPTLOCHD =RPTOBJ RPTOBJHD
QTY	=ENTRY RPTLOC RPTOBJ
rr	=CHPARTY =CONT =LOCATION =RPTLOC =RPTLOCHD =RPTOBJ RPTOBJHD =SECTOR
rrala	=SECTOR
rralb	=SECTOR
rralc	=SECTOR
rrald	=SECTOR
rrc	=CONT
rrd	=LOCATION
rrp	=LOCATION
rrv	=LOCATION
rtaidamt	=RPTOBJ
rtfafsbene	=RPTOBJ
rtindbene	=RPTOBJ
rtmafsbene	=RPTOBJ
rtnafsbene	=RPTOBJ
rttranspor	=RPTOBJ
rtunsbene	=RPTOBJ
screen	RPTLOC RPTOBJ
SEQ	=ENTRY REBUILD
setdata	=RPTLOC =RPTOBJ
SUBGRANT	=ENTRY REBUILD
times	=ENTRY
TRANSPORT	=ENTRY RPTLOC RPTOBJ

VARIABLES: GLOBAL CONCORDANCE (continued)

ariable occurs in	Modules (modified variables indicated by '=')
NSBENE	=ENTRY RPTLOC RPTOBJ
PDATE	=ENTRY
aidamt	=ENTRY
begin	=ENTRY
choose	=LOCATION
conabbrev	=CONT =ENTRY
donated	=CHPARTY =ENTRY
dpi	=ENTRY
end	=ENTRY
ERDATE	=ENTRY RPTLOC RPTOBJ
ERSTAT	=ENTRY RPTLOC RPTOBJ
fafsbene	=CHPARTY =ENTRY
indbene	=CHPARTY =ENTRY
intermed	=ENTRY
mafsbene	=CHPARTY =ENTRY
nafsbene	=CHPARTY =ENTRY
party	=CHPARTY =ENTRY
partyno	=CHPARTY =ENTRY
price	=ENTRY
qtr	=ENTRY
qty	=ENTRY
seq	=ENTRY
subgrant	=ENTRY
transport	=ENTRY
unsbene	=CHPARTY =ENTRY
update	=ENTRY
verdate	=ENTRY
verstat	=ENTRY
WAFGHS	CHPARTY ENTRY
WAMT	ENTRY
WDATES	ENTRY
WDPI	ENTRY
WINDIR	CHPARTY ENTRY
WNAFGHS	CHPARTY ENTRY
WTRNSP	ENTRY
WUNSAL	CHPARTY ENTRY
WVERIF	ENTRY
year	=ENTRY
w	ENTRY SECTOR
YEAR	=ENTRY REBUILD RPTLOC RPTOBJ
yearset	=RPTLOC RPTLOCHD =RPTOBJ RPTOBJHD

DATA FILES USED IN SYSTEM REP

DBF FILE ALA

No Aliases

Opened in: ENTRY REBUILD RPTLOC RPTOBJ

Indexes

Ix	Index file	Index expression
6	ALA	ALA

Fields

Field	Length	Type	Ix	Picture	Picture taken from
ALA	1	C	6		
ALANAME	31	C			
Total	32				

FIELDS CONCORDANCE FOR DBF FILE ALA

Field	occurs in Modules				
ALA	=ENTRY	REBUILD	RPTLOC	RPTOBJ	SECTOR
ALANAME	RPTLOC	RPTOBJ	SECTOR		

DATA FILES USED IN SYSTEM REP

DBF FILE ALB

No Aliases

Opened in: ENTRY REBUILD RPTLOC RPTOBJ

Indexes

Ix	Index file	Index expression
7	ALB	ALA+ALB

Fields

Field	Length	Type	Ix	Picture	Picture taken from
ALA	1	C	7		
ALB	1	C	7		
ALBNAME	32	C			
Total	34				

FIELDS CONCORDANCE FOR DBF FILE ALB

Field	occurs in Modules				
ALA	=ENTRY	REBUILD	RPTLOC	RPTOBJ	SECTOR
ALB	=ENTRY	REBUILD	RPTLOC	RPTOBJ	SECTOR
ALBNAME	RPTLOC	RPTOBJ	SECTOR		

DATA F: :S USED IN SYSTEM REP

DBF FILE ALC

No ..liases

Opened in: ENTRY REBUILD RPTLC'C RPTOBJ

Indexes

Ix	Index file	Index expression
8	ALC	ALA+ALB+ALC

Fields

Field	Length	Type	Ix	Picture	Picture taken from
ALA	1	C	8		
ALB	1	C	8		
ALC	1	C	8		
ALCNAME	31	C			
Total	34				

FIELDS CONCORDANCE FOR DBF FILE ALC

Field	occurs in Modules				
ALA	=ENTRY	REBUILD	RPTLOC	RPTOBJ	SECTOR
ALB	=ENTRY	REBUILD	RPTLOC	RPTOBJ	SECTOR
ALC	=ENTRY	REBUILD	RPTLOC	RPTOBJ	SECTOR
ALCNAME	RPTLOC	RPTOBJ	SECTOR		

DATA FILES USED IN SYSTEM REP

DBF FILE ALD

o Aliases

pened in: ENTRY REBUILD RPTLOC RPTOBJ

Indexes

Ix	Index file	Index expression
9	ALDNAME	ALA+ALB+ALC+ALDname
10	ALD	ALA+ALB+ALC+ALD

Fields

ield	Length	Type	Ix	Picture	Picture taken from
LA	1	C	10		
LB	1	C	10		
LC	1	C	10		
LD	2	C	10		
LDNAME	30	C	9		
EAS	14	C			
RICEUNIT	14	C			
NDIC	5	C			
WDPI	1	L			
WMEAS	1	L			
WAMT	1	L			
WTRNSP	1	L			
WAFGHS	1	L			
WNAFGHS	1	L			
WUNSAL	1	L			
WINDIR	1	L			
WDATES	1	L			
WVERIF	1	L			
Total	78				

FIELDS CONCORDANCE FOR DBF FILE ALD

Field occurs in	Modules
ALA	=ENTRY REBUILD RPTLOC RPTOBJ SECTOR
ALB	=ENTRY REBUILD RPTLOC RPTOBJ SECTOR
ALC	=ENTRY REBUILD RPTLOC RPTOBJ SECTOR
ALD	=ENTRY REBUILD RPTLOC RPTOBJ SECTOR
ALDNAME	REBUILD RPTLOC RPTOBJ SECTOR
INDIC	CHPARTY ENTRY RPTLOC RPTOBJ
MEAS	ENTRY RPTLOC RPTOBJ
PRICEUNIT	ENTRY RPTLOC RPTOBJ
VWAFGHS	CHPARTY ENTRY
VWAMT	ENTRY
VWDATES	ENTRY
VWDPI	ENTRY
VWINDIR	CHPARTY ENTRY
VWNAFGHS	CHPARTY ENTRY
VWTRNSP	ENTRY

FIELDS CONCORDANCE FOR DBF FILE ALD (continued)

Field	occurs in Modules	
VWUNSAL	CHPARTY	ENTRY
VWVERIF	ENTRY	

DATA FILES USED IN SYSTEM REP

DBF FILE CONTRACT

Aliases

pened in: DEFAULT ENTRY REBUILD

Indexes

Ix	Index file	Index expression
11	CONNAME	conname
12	CONNO	conno

Fields

ield	Length	Type	Ix	Picture	Picture taken from
ONNO	2	C	12		
ONABBREV	13	C			
ONNAME	42	C	11		
Total	57				

FIELDS CONCORDANCE FOR DBF FILE CONTRACT

ield	occurs in Modules
ONABBREV	CONT DEFAULT =ENTRY REBUILD RPT OC RPTOBJ
ONNAME	CONT DEFAULT REBUILD
ONNO	CONT =ENTRY REBUILD

DATA FILES USED IN SYSTEM REP

DBF FILE DISTRICT

No Aliases

Opened in: ENTRY REBUILD RPTLOC RPTOBJ

Indexes

Ix	Index file	Index expression
2	PDISTRICT	provno+district
3	PDISTNO	provno+dis+no

Fields

Field	Length	Type	Ix	Picture	Picture taken from
PROVNO	2	C	3		
DISTNO	4	C	3		
DISTRICT	25	C	2		
Total	31				

FIELDS CONCORDANCE FOR DBF FILE DISTRICT

Field occurs in	Modules
DISTNO	=ENTRY LOCATION REBUILD RPTLOC RPTOBJ
DISTRICT	LOCATION REBUILD RPTLOC RPTLOCHD RPTOBJ
PROVNO	=ENTRY LOCATION REBUILD RPTLOC RPTLOCHD RPTOBJ

DATA FILES USED IN SYSTEM REP

DBF FILE INDIC

No Aliases

Opened in: RPTLOC RPTOBJ

Indexes

Ix	Index file	Index expression
16	INDNO	INDNO

Fields

Field	Length	Type	Ix	Picture	Picture taken from
INDNO	1	C	16		
OBJECTIVE	60	C			
INDICATOR	60	C			
COMMENT	60	C			
Total	181				

FIELDS CONCORDANCE FOR DBF FILE INDIC

Field	occurs in Modules	
INDICATOR	RPTLOC	RPTOBJHD
INDNO	RPTLOC	RPTOBJ
OBJECTIVE	RPTLOC	RPTOBJHD

DATA FILES USED IN SYSTEM REP

DBF FILE NEWPLACE

No Aliases

Opened in: ENTRY

NEWPLACE has no evidently related index files.

Field	Length	Fields Type	Ix	Picture	Picture taken from
PROVNO	2	C			
DISTNO	4	C			
PLACE	40	C			
MENTIONED	4	N		9999	
Total	50				

FIELDS CONCORDANCE FOR DBF FILE NEWPLACE

Field	occurs in Modules
DISTNO	=ENTRY LOCATION REBUILD RPTLOC RPTOBJ
PLACE	LOCATION REBUILD
PROVNO	=ENTRY LOCATION REBUILD RPTLOC RPTLOCHD RPTOBJ

DATA FILES USED IN SYSTEM REP

DBF FILE PARTY

No Aliases

Opened in: ENTRY REBUILD

Indexes

Ix	Index file	Index expression
14	PARTY	party
15	PARTYNO	partyno

Fields

Field	Length	Type	Ix	Picture	Picture taken from
PARTYNO	2	C	15		
PARTY	28	C	14		
Total	30				

FIELDS CONCORDANCE FOR DBF FILE PARTY

Field	occurs in Modules
PARTY	CHPARTY ENTRY REBUILD
PARTYNO	CHPARTY =ENTRY REBUILD

DATA FILES USED IN SYSTEM REP

DBF FILE PLACE

No Aliases

Opened in: ENTRY REBUILD

Indexes

Ix	Index file	Index expression
4	PDPLACE	provno+distno+place
5	PDPLACENO	provno+distno+placeno
19	PDPLACEN	provno+distno+placeno

Fields

Field	Length	Type	Ix	Picture	Picture taken from
PROVNO	2	C	19		
DISTNO	4	C	19		
PLACENO	3	C	19		
PLACE	40	C	4		
Total	49				

FIELDS CONCORDANCE FOR DBF FILE PLACE

Field	occurs in Modules					
DISTNO	=ENTRY	LOCATION	REBUILD	RPTLOC	RPTOBJ	
PLACE	LOCATION	REBUILD				
PLACENO	=ENTRY	LOCATION	REBUILD			
PROVNO	=ENTRY	LOCATION	REBUILD	RPTLOC	RPTLOCHD	RPTOBJ

DATA FILES USED IN SYSTEM REP

DBF FILE PROVINCE

No Aliases

Opened in: ENTRY REBUILD RPTLOC RPTOBJ

Indexes

Ix	Index file	Index expression
1	PROVNO	provno
17	PROVINCE	province

Fields

Field	Length	Type	Ix	Picture	Picture taken from
PROVNO	2	C	1		
PROVINCE	25	C	17		
Total	27				

FIELDS CONCORDANCE FOR DBF FILE PROVINCE

Field	occurs in Modules
PROVINCE	LOCATION REBUILD RPTLOCHD RPTOBJ
PROVNO	=ENTRY LOCATION REBUILD RPTLOC RPTLOCHD RPTOBJ

DATA FILES USED IN SYSTEM REP

DBF FILE REPDATA

No Aliases

Opened in: ENTRY REBUILD RPTLOC RPTOBJ UTIL

Indexes

Ix	Index file	Index expression
13	REPDATA	qtr+year+provno+distno+placeno+ala+alb+alc+ald+conno+subrant+seq
18	RPT1	provno+distno+ala+alb+alc+ald+conabbrev

Fields

Field	Length	Type	Ix	Picture	Picture taken from
QTR	1	C	13		
YEAR	2	C	13		
PROVNO	2	C	18		
DISTNO	4	C	18		
PLACENO	3	C	13		
ALA	1	C	18		
ALB	1	C	18		
ALC	1	C	18		
ALD	2	C	18		
CONNO	2	C	13		
CONABBREV	13	C	18		
SEQ	2	C	13		
SUBGRANT	2	C	13		
QTY	4	N		9999	RPTOBJ
AIDAMT	8	N		999999999	RPTOBJ
TRANSPORT	8	N		9999999	RPTOBJ
MAFSBENE	6	N		999999	RPTOBJ
FAFSBENE	6	N		999999	RPTOBJ
NAFSBENE	6	N		999999	RPTOBJ
UNSBENE	8	N		9999999	RPTOBJ
INDBENE	8	N		9999999	RPTOBJ
DPI	1	L			
DONATED	1	L			
PRICE	1	L			
BEGIN	8	D			
END	8	D			
VERSTAT	2	N		99	
VERDATE	8	D			
PARTYNO	2	C			
INTERMED	30	C			
UPDATE	8	D			
Total	159				

FIELDS CONCORDANCE FOR DBF FILE REPDATA

Field	occurs in Modules
AIDAMT	=ENTRY RPTLOC RPTOBJ
ALA	=ENTRY REBUILD RPTLOC RPTOBJ SECTOR
ALB	=ENTRY REBUILD RPTLOC RPTOBJ SECTOR
ALC	=ENTRY REBUILD RPTLOC RPTOBJ SECTOR
ALD	=ENTRY REBUILD RPTLOC RPTOBJ SECTOR
BEGIN	=ENTRY RPTLOC RPTOBJ
CONABBREV	CONT DEFAULT =ENTRY REBUILD RPTLOC RPTOBJ
CONNO	CONT =ENTRY REBUILD
DISTNO	=ENTRY LOCATION REBUILD RPTLOC RPTOBJ
DONATED	=ENTRY RPTLOC RPTOBJ
DPI	=ENTRY RPTLOC RPTOBJ
END	=ENTRY RPTLOC RPTOBJ
FAFSBENE	=ENTRY RPTLOC RPTOBJ
INDBENE	=ENTRY RPTLOC RPTOBJ
INTERMED	=ENTRY
MAFSBENE	=ENTRY RPTLOC RPTOBJ
NAFSBENE	=ENTRY RPTLOC RPTOBJ
PARTYNO	CHPARTY =ENTRY REBUILD
PLACENO	=ENTRY LOCATION REBUILD
PRICE	=ENTRY RPTLOC RPTOBJ
PROVNO	=ENTRY LOCATION REBUILD RPTLOC RPTLOCHD RPTOBJ
QTR	=ENTRY REBUILD RPTLOC RPTOBJ
QTY	=ENTRY RPTLOC RPTOBJ
SEQ	=ENTRY REBUILD
SUBGRANT	=ENTRY REBUILD
TRANSPORT	=ENTRY RPTLOC RPTOBJ
UNSBENE	=ENTRY RPTLOC RPTOBJ
UPDATE	=ENTRY
VERDATE	=ENTRY RPTLOC RPTOBJ
VERSTAT	=ENTRY RPTLOC RPTOBJ
YEAR	=ENTRY REBUILD RPTLOC RPTOBJ

Recursion check: OK (No circular references)

HIERARCHY TREE FOR SYSTEM
REP

```
EP
 ----DEFAULT
 ----ENTRY
     |----CHPARTY
     |----CONT
     |----LOCATION
     `----SECTOR
 |----RPTLOC
 |   `----RPTLOCHD
 |----RPTOBJ
 |   `----RPTOBJHD
 `----UTIL
     `----REBUILD
```

Cross-reference for system REP follows.

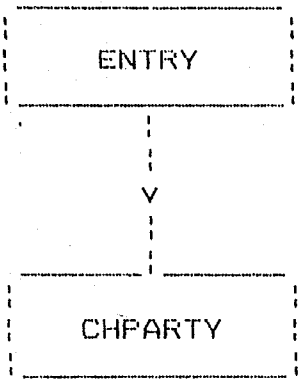

Diagram for module CHPARTY
in system REP

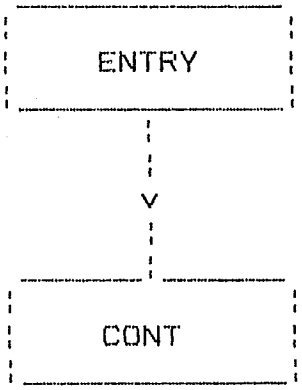

Diagram for module CONT
in system REP

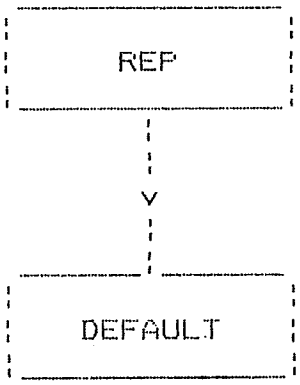

Diagram for module DEFAULT
in system REP

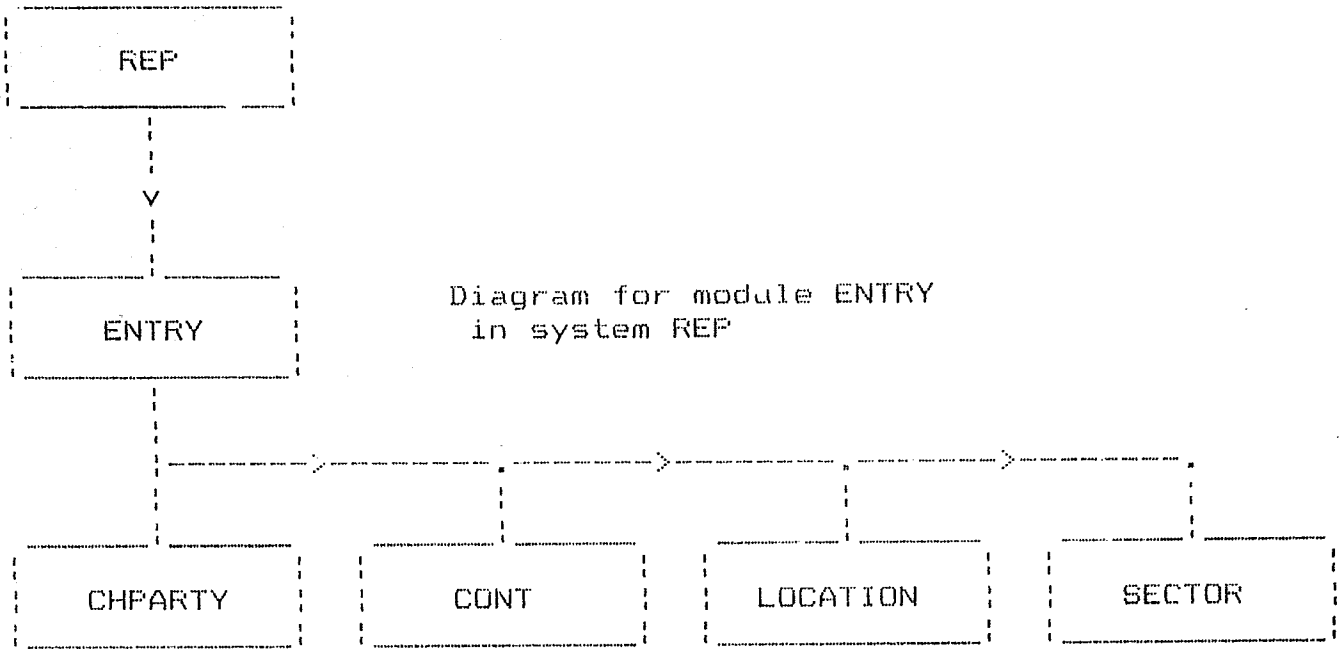

Diagram for module ENTRY
in system REP

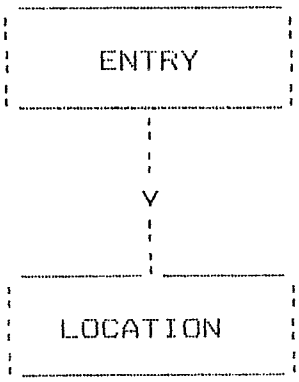

Diagram for module LOCATION
in system REP

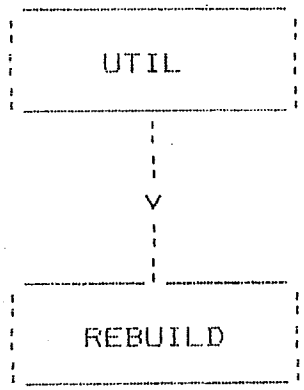

Diagram for module REBUILD
in system REP

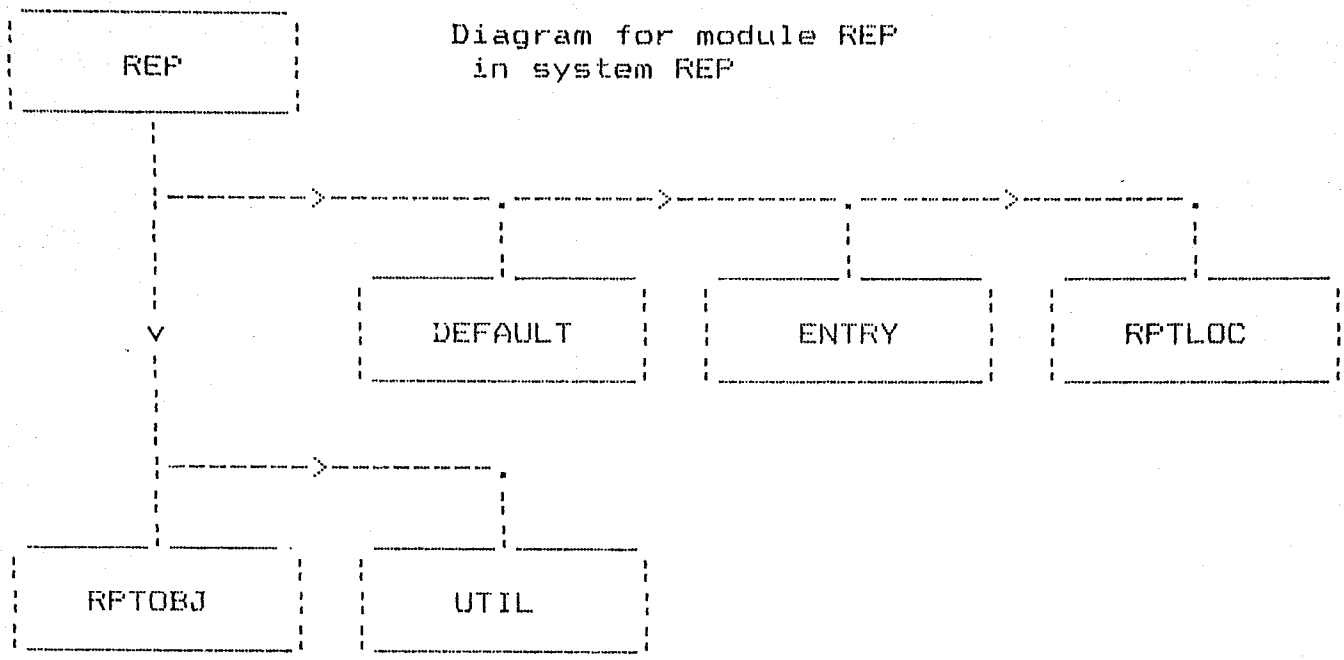

Diagram for module REP
in system REP

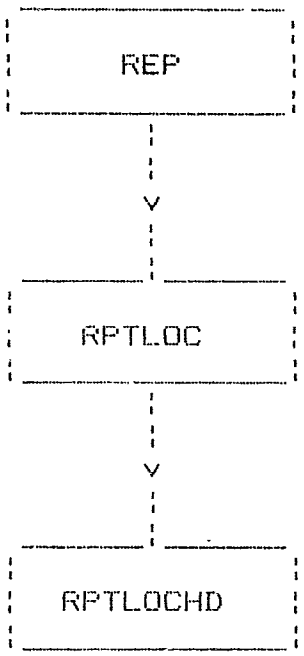

Diagram for module RPTLOC
in system REP

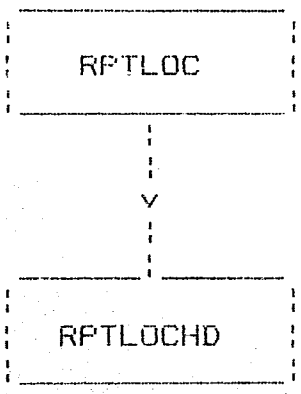

Diagram for module RPTLOCHD
in system REP

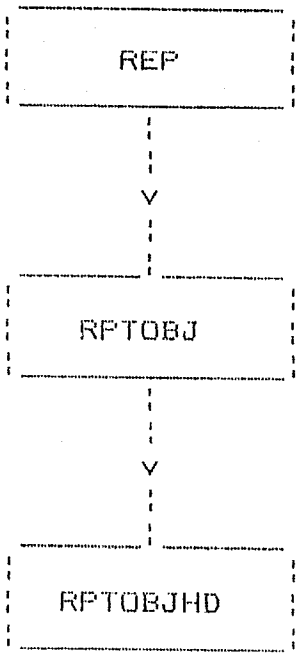

Diagram for module RPTOBJ
in system REP

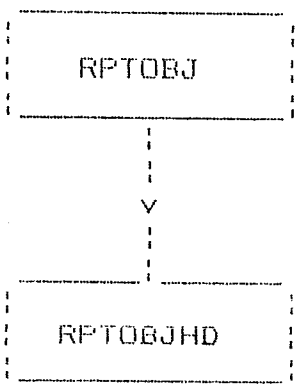

Diagram for module RPTOBJHD
in system REP

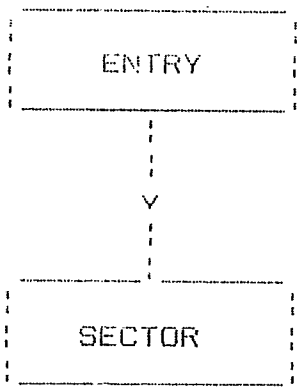

Diagram for module SECTOR
in system REP

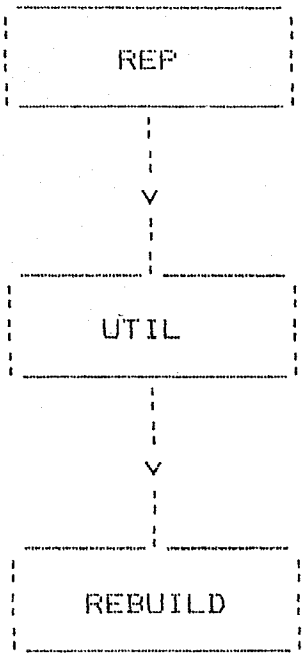

Diagram for module UTIL
in system REP

** END OF ANALYSIS **