

I-1. arcmenu

1.0 General Information

The arcmenu program creates a menu system that allows access to most of the programs available in the RFC Archive System (RAX). It is a simple text based menu program written in C. This application can be run by any user.

1.1 Enhancements/Bug Fixes/Changes

Build OB7.2

This application was recompiled because of the upgrade of the OS in ob7.2. In addition, the menu files were updated to reflect changes in applications due to the move from Informix to Postgres.

Build OB6

Enhancements

- Added new shefdecoders menu with five options
- Added new option to the main menu for arcnave application

Build OB4

Enhancements

- Added new admin menu with five options
- Added new option to the data processing menu

2.0 Configuration Files

The menu system for the archive db system was designed to be simple and configurable. When the user types in the name of the menu system, i.e. arcmenu, the program searches the apps_defaults system to find the directory where the menu files are stored. Currently there are five menu files and these files can be found on the RAX in the directory /rfc_arc/cfg/menu. The files are called arc.menu, dataproc.menu, otherdb.menu, shef.menu and admin.menu.

3.0 User How-To

The user accesses the arcmenu program with the following steps.


```
-----> Other Viewing/Extraction <-----  
  
1. forms -- invoke adbpg.pl script for DB table access  
2. dcextract -- extract data and format into DATACARD fmt  
  
M. Main menu           S. SHEF Encoding menu  
P. Data Processing menu A. Admin Menu  
Q. quit                D. Shefdecoders menu  
  
ENTER Selection:
```

Figure 2. Other Viewing/Extraction Menu

```
-----> SHEF Encoding <-----  
  
1. dcparse -- output SHEF from DATACARD  
2. nrscdlyparse -- output SHEF from NRCS historical fmt  
3. usgsdlyparse -- output SHEF from USGS daily fmt  
  
M. Main menu           V. Other Viewing/Extraction menu  
P. Data Processing menu A. Admin menu  
Q. quit                D. Shefdecoders menu  
  
ENTER Selection:  
ENTER Selection: p
```

Figure 3. SHEF Encoding Menu

```
-----> Data Processing <-----  
  
1. process_precip -- process precip (PC -> PP)  
2. process_stage -- process stage (height -> flow,storage)  
3. process_temp -- process temperature (TA -> TA,TX,TN)  
4. process_sw -- process swe  
5. process_flow -- process flow  
  
6. transfer_txn -- (TX,TN -> TX,TN)  
7. transfer_precip -- (PP -> PP)  
  
8. slope_to_stage == translate slope measurements to stages  
  
M. Main menu  
V. Other Viewing/Extraction menu  
Q. quit  
  
S. SHEF Encoding menu  
A. Admin menu  
D. Shefdecoders menu  
  
ENTER Selection:
```

Figure 4. Data Processing Menu

```
-----> DB & System Admin <-----  
  
1. find_bigfiles -- finds files > 1.1mb in selected directories  
2. rax_status_df -- view most recent message created by the cron  
3. purge_files -- this routine deletes selected old log files  
4. run_vacuumdb -- run the postgres vacuumdb command  
5. run_pgbkups -- allows user to run pg_dump manually  
  
M. Main menu  
P. Data Processing menu  
Q. quit  
  
V. Other Viewing/Extraction menu  
S. SHEF Encoding menu  
D. Shefdecoders menu  
  
ENTER Selection:
```

Figure 5. DB & System Admin Menu

```
-----> shefdecoders <-----  
  
  *** ALWAYS CHECK THE STATUS BEFORE ***  
  *** STARTING ONE OF THE DECODERS   ***  
  
  1.  status of shefdecoders  -- are they running  
  
  2.  start_raw_decoder  
  
  3.  start_processed_decoder  
  
  4.  stop_raw_decoder  
  
  5.  stop_processed_decoder  
  
  M.  Main menu                V.  Other Viewing/Extraction menu  
  P.  Data Processing menu     A.  Admin menu  
  Q.  quit                    S.  SHEF Encoding menu  
  
ENTER Selection:
```

Figure 6. Shefdecoders menu

For this submenu, a new script called `status_decoders` was created. This is the script that is run by option 1. The script is located in the directory `/rfc_arc/scripts/decoders` and simply executes the command “`ps -ef | grep shef`”.

4.0 Editing the Menu Files

The menu system for the archive db system was designed to be simple and configurable. When the user types in the name of the menu system, i.e. `arcmenu`, the program searches the `apps-defaults` system to find the directory where the menu files are stored. The initial menu is called `arc.menu`. Here is an earlier version of the `/rfc-arc/cfg/menu/arc.menu` file:

```
# this is the first menu for the menu program  
# the ! in the first column specifies a menu instruction  
# fields are separated by the | (pipe) symbol  
# the 1st field is the menu choice  
# the 2nd field is the program name  
#   - a full pathname may be used if desired  
# the 3rd field is command line arguments for the program  
# the 4th field is a wait indicator - so the menu will ask for  
#   - a CR after the program has completed and before clearing the screen.  
#  
# the listmenu command lists another menu, spacing is critical for this,  
#   see implementation below  
#  
# every line without a ! or # in the first column is printed
```

```
#      to stdout
# the ! lines can be anywhere in the file (e.g. grouped at the bottom)
```

```

/-----/
/-----/
|
|           Archive Database
|           Programs
|
|-----|
|-----|
/-----/

```

```
1.  DataView -- view and edit archive data
```

```
2.  ff_oper_view -- flat file viewer
```

```
3.  Display_rc -- rating curve viewer
```

```
V.  Other Viewing/Extraction menu
```

```
S.  SHEF Encoding menu
```

```
P.  Data Processing menu
```

```
Q   quit
```

```
#
#
! 1 | datview | | |
! 2 | /rfc_arc/bin/fam/ff_oper_view.tcl | | |
! 3 | display_rc | | |
! q | exit |
! v | listmenu otherdb.menu|
! p | listmenu dataproc.menu|
! s | listmenu shef.menu|
! m | listmenu arc.menu|
```

The menu program will read this file and write to stdout any line that does not begin with an exclamation point or a pound sign. Any line that begins with an exclamation point is a menu instruction consisting of four fields separated by the pipe symbol. The 1st field is the menu choice, the 2nd field is the program name (a full pathname may be used if desired), the 3rd field is command line arguments for the program, and the 4th field is a wait indicator (so the menu will ask for a carriage return (CR) after the program has completed and before clearing the screen). The listmenu command in the 2nd field is a special instruction to the menu program that directs it to list another menu. Sub-menus may be implemented in this fashion, and there is no limit to the levels of sub-menus you may have. Here is a sample sub-menu file:

```
-----> Data Processing <-----
1.  process_precip -- process precip (PC -> PP)
2.  process_stage -- process stage (height -> flow,storage)
3.  process_temp -- process temperature (TA -> TA,TX,TN)
4.  process_sw -- process swe
5.  process_flow -- process flow

6.  transfer_txn -- (TX,TN -> TX,TN)
7.  transfer_precip -- (PP -> PP)
```

M. Main menu S. SHEF Encoding menu
V. Other Viewing/Extraction menu

Q. quit

```
#  
#  
! 1 | process_precip | | |  
! 2 | process_stage | | |  
! 3 | process_temp | | |  
! 4 | process_sw | | |  
! 5 | process_flow | | |  
! 6 | transfer_txn | | |  
! 7 | transfer_precip | | |  
! m | listmenu arc.menu |  
! v | listmenu otherdb.menu |  
! s | listmenu shef.menu |  
! q | exit |
```

Note that entry points are defined to return to the main menu or to list other menus. Also, the pathname of the program may be omitted in a menu config file if the path is in the user's PATH env variable.

5.0 Troubleshooting Information

If for some reason the application fails, contact the RFC Support Group.