

## Session 9: Menu Programs

At the end of this session participants will be able to:

- Understand the how to design the screens of a menu program
- Use the command `execpff()` to launch one CPro application from another.
- Create a simple menu program to launch the main data entry program.

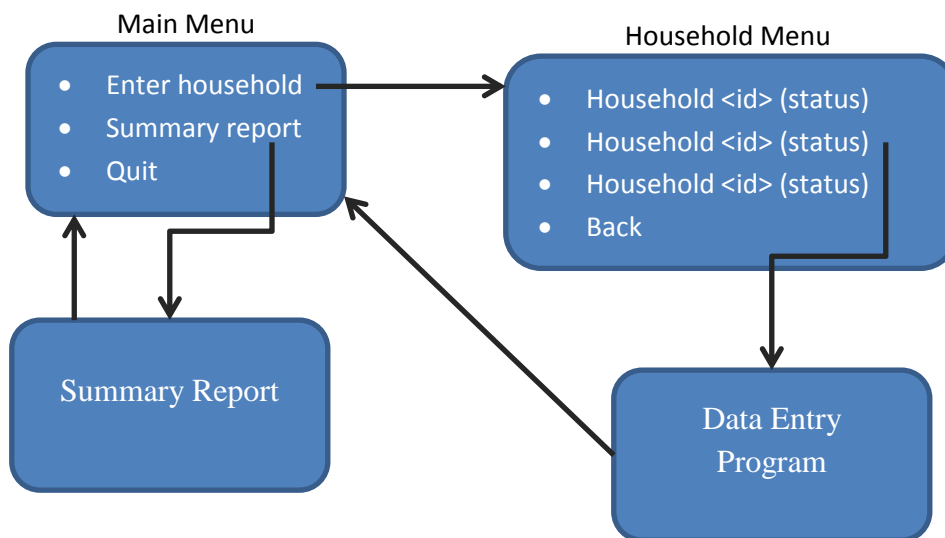
### What is a Menu Program

A menu program is a CPro data entry application that is used to manage the data entry workflow. A menu program is not used for capturing any interview data itself. Instead it launches other data entry programs for interviews. Menu programs generally have some or all of following functions:

- Launches other CPro applications to do data collection (often pre-filling id-items)
- Show reports on progress, summary statistics
- Manage user access through userid/passwords
- Manage household listing and interview assignments
- Launch synchronization

### Designing the Screens and Flow

Once you decide on which functions you want your menu program to perform the next step is to design the screens and how they link together. This is most easily done as a diagram like the one below for a fairly simple menu program.



## Creating the Dictionary and Forms

Once you have the screens drawn out the next step is to create a new data entry application for your menu program. Each different menu screen is will be a variable in the dictionary. The value set for the variable will show the available menu choices to the enumerator on that screen. In the menu example above we will have two variables: MAIN\_MENU (value set: Enter Household - 1, Summary Report – 2, Quit-9) and HOUSEHOLD\_MENU with a dynamic value set.

Unlike typical data entry applications, menu programs tend not to have a linear flow. As a result the order of the variables on the form is less important. We will use skips and reenters to move from one menu to another.

## Menu Program Logic

The logic for processing the menu choice for each screen goes in the postproc of the variable for the menu. For example to process the main menu field in our example we would have the following logic:

```
PROC MAIN_MENU
postproc

// Handle the menu choice
if MAIN_MENU = 1 then
    // Add household
    skip to HOUSEHOLD_MENU;
elseif MAIN_MENU= 2 then
    // Summary report
    skip to SUMMARY_REPORT;
elseif MAIN_MENU= 9 then
    // Quit
    stop(1);
endif;
```

The logic for handling for other fields would be similar with options such as “Back” using a reenter to return to the main menu.

## Launching one CPro Application from Another

One key function of a menu program is to launch a second data entry application. We can launch other CPro programs from within our data entry program using *execpff()*. You give *execpff* the path to a pff file for a CPro application and it will start it. In addition to the filename you can also pass it either the keyword *wait* or the keyword *stop*. Using *wait* will pause the menu application until the application that was launched exits while using *stop* will exit the menu application immediately after launching the other application. For example the following logic would launch a data entry application using the pff file *Entry.pff* in the sibling directory to the directory that menu program is in.

```
execpff("../DataEntry/Entry.pff", stop);
```

This will start the data entry application and immediately exit the menu. The “..” in the path to Entry.pff tells CSEntry to go one directory up, i.e. to go to the parent of the Menu directory and from there to DataEntry/Entry.pff.

We can modify the pff file so that it automatically starts in add mode. Right click on Entry.pff and choose “Edit” to bring up the pff file editor. From there we can choose the start mode, name of the data file, external files and a host of other parameters.

More advanced menu programs will write out the pff file before executing it in order to be able to pre-fill in the id-items and modify the data file name.

## Making the Entry Program Return to the Menu Program

Normally if we start a CSEntry application in add mode and complete a questionnaire, CSEntry will automatically begin entry of the next questionnaire. When starting from the menu program we do not want this behavior. Instead we want to return to the menu program as soon as we finish the questionnaire. We can do this by adding a call to *execpff()* to relaunch the menu program. Which proc can we put this in so that it will be called when we are done with a questionnaire? The postproc of the level is that last proc to be called for each questionnaire so we can put it there.

```
PROC ENTRY_QUESTION  
  
// If launched by menu we should stop and return to menu  
// here after questionnaire is completed. If not, in add mode  
// we will start to add another household instead of going  
// back to the menu.  
execpff("../Menu/Menu.pff", stop);
```

We also need to handle the case where the interviewer uses the stop button on Windows or the back button on Android to exit the data entry application. To do this we provide an OnStop function the entry application logic that relaunches the menu program. OnStop is a special function. If CSPro finds a function named OnStop in your logic it will call it when the interviewer pushes the stop button (or the back button on Android).

```
function OnStop()  
    execpff("../Menu/Menu.pff", stop);  
end;
```

## Deploying multiple applications on Android

When you copy your application to Android it is important that you preserve the same folder structure as you have on the PC. In our case we must create a separate menu folder to contain the .pen and .pff files for the menu and this menu folder must be in the same parent directory as the DataEntry folder so that when we use "../DataEntry/" from the menu application it points to the folder containing the entry application we are launching.

One way to simplify deployment is to use a Windows batch file to generate the .pen files for both the entry application and the menu application and copy them into a deployment folder that can be copied onto the device. This can greatly speed up testing of your application on Android.

```
setlocal

REM Find CSEntry.exe (path differs on 32 and 64 bit Windows)

SET CSEntry="%ProgramFiles(x86)%\CSPro 6.1\CSEntry.exe"
if exist %CSEntry% goto :gotcspro
SET CSEntry="%ProgramFiles%\CSPro 6.1\CSEntry.exe"
if exist %CSEntry% goto :gotcspro
echo "Can't find CSEntry version 6.1. Is it installed?"
goto :eof
:gotcspro

REM Create deployment directory
rmdir /q /s Deployment
mkdir Deployment
cd Deployment
mkdir DataEntry
mkdir Menu
cd..

REM Create .pen files
%CSEntry% /pen .\Menu\Menu.ent
%CSEntry% /pen .\DataEntry\Entry.ent

REM Copy applications to deployment
move /y .\Menu\Menu.pen .\Deployment\Menu
move /y .\Entry\ DataEntry.pen .\Deployment\DataEntry

REM Copy .pff for Menu
copy /y .\Menu\Menu.pff .\Deployment\Menu\Menu.pff

REM Copy Lookup files if any
copy /y .\DataEntry\VillagesPerDistrict.dat .\Deployment\ DataEntry \
```