

## Session 7: Multimedia & GPS

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

- Use images in value sets
- Launch external applications from a data entry application
- Launch photo and document viewers from a data entry application
- Capture GPS coordinates in a data entry application
- Display captured GPS points on a map
- Generate and view reports from a data entry application

### Value Set Images

In addition to showing text in a value set we can also display images. To add an image to a value set, select the variable in the dictionary editor and click on the “...” button next to the value. Browse to pick an image file. CSPro supports standard image file formats: png, jpeg, bmp... We recommend using png or jpeg as they tend have smaller file sizes. The images are not added to the pen file by default so it is a good idea to first put your images in a resources directory so that you won't have to worry about copying them to the device.

Let's add images for the roof types in F07. First copy the RoofImages folder into the resource directory. Now open the value set for F07 and add each image to its respective value. We do not have a picture for other (specify) so we will leave that one without an image. Run the application on Windows and on Android and see the images.

### Launching External Applications on Windows

The command **execsystem()** lets you launch another application from logic. It takes the name of the application to launch. For example, using a userbar button you could bring up the Window's calculator:

```
PROC GLOBAL

// Launch the calculator
function showCalculator()
    execsystem("calc.exe");
end;

PROC SURVEY_FF
preproc
userbar(clear);
userbar(add button, "Calculator", showCalculator);
userbar(show);
```

For Windows utilities like Calculator and Notepad which are in the system directory you can simply give the name. For other applications that are installed in the Program Files directory you may need to specify the full path. For example, to launch Microsoft Word (Office 2016 version) you would use:

```
execsystem("C:\Program Files (x86)\Microsoft Office\root\Office16\WINWORD.EXE");
```

## Launching External Applications on Android

The `execsystem()` command is slightly different on Android. On Android to launch an application with `execsystem()` use “app:” followed by the package name of the application. For example, to launch Gmail:

```
execsystem("app:com.google.android.gm");
```

Finding the package name can be a bit tricky. The easiest way is to use your web browser to search for the application on the Google Play website ([play.google.com](http://play.google.com)). On the page for the application the package name will be the last part of the address in the address bar of your browser, after the “?id=”. For example, if you go the page for CSEntry on the Google Play Store the address is:

```
https://play.google.com/store/apps/details?id=gov.census.cspro.csenry
```

The package name is therefore “*gov.census.cspro.csenry*”.

Going back to launching the calculator from Android, unfortunately different Android devices have different calculators so you have try multiple package names. The following code tries the two different versions that we have seen:

```
// Samsung uses app:com.sec.android.app.popupcalculator,  
// LG and HTC use app:com.android.calculator2.  
// We will try one and if it doesn't work then try the other.  
if execsystem("app:com.sec.android.app.popupcalculator") <> 1 then  
    if execsystem("app:com.android.calculator2") <> 1 then  
        errmsg("Unable to launch calculator.");  
    endif;  
endif;
```

To make our code work on both Android and Windows we can use the function `getos()` which returns a number between 10 and 19 for Windows and between 20 and 29 for Android.

```
// Launch the calculator  
function showCalculator()  
    if getos() in 10:19 then  
        // Windows  
        execsystem("calc.exe");  
    else  
        // Android - different devices have different built in calculators.  
        // Samsung uses app:com.sec.android.app.popupcalculator,  
        // LG and HTC use app:com.android.calculator2.  
        // We will try one and if it doesn't work then try the other.  
        if execsystem("app:com.sec.android.app.popupcalculator") <> 1 then  
            if execsystem("app:com.android.calculator2") <> 1 then
```

```
                errmsg("Unable to launch calculator.");
            endif;
        endif;
    endif;
end;
```

## Viewing Files on Android

On Android `execsystem()` can also be used to launch files with their default viewer.

```
execsystem("view:/mnt/sdcard/csentry/picture.jpg");
execsystem("view:/mnt/sdcard/csentry/audio.mp3");
execsystem("view:/mnt/sdcard/csentry/movie.mp4");
execsystem("view:/mnt/sdcard/csentry/document.pdf");
```

For this to work, the device must have an application installed capable of viewing the type of file specified. Android determines which application to launch to view the file based on the file extension. All Android devices have viewers for most photo, music, and video file formats, however not all have a built-in PDF viewer. If you do not have one on your device, you can always download one from Google Play.

Let's add a userbar button to show a map of the enumeration area as a visual aid for the interviewer.

```
// Display an image containing a map of the enumeration area
function showEAMap()
    execsystem("view:/mnt/sdcard/csentry/Popstan2020/Housing/resources/EAMap.png");
end;
```

We can make our code a little more flexible by using the function `pathname()` to get the directory where the application is stored. This way if someone copies it into a different folder on the phone/tablet it will still work:

```
// Display an image containing a map of the enumeration area
function showEAMap()
    execsystem(makertext("view:%s/resources/EAMap.png",
                        pathname(Application)));
end;
```

Our code is currently limited to a single enumeration area. We could improve it to show a different image based on the current value of the enumeration area variable.

```
// Display an image containing a map of the enumeration area
function showEAMap()
    execsystem(makertext("view:%s/resources/EAMap%d.png",
        pathname(Application), ENUMERATION_AREA));
end;
```

## Taking Photos on Android

To take a picture on Android you can use *execsystem* with “camera:” followed by the full path in which to save the photo:

```
execsystem("camera:/mnt/sdcard/csentry/photo.jpg");
```

Note that photos do not get saved into the application data file. You need to save them to a folder/file on the device that will make it easy to link the photo back to the interview that it was taken for. The easiest way to do this is to make the name of the photo based on the id-items of the questionnaire:

```
string photoFilename = makertext("photo%d%02d%03d%d%03d.jpg",
    PROVINCE, DISTRICT, ENUMERATION_AREA,
    AREA_TYPE, HOUSEHOLD_NUMBER);
execsystem(makertext("camera:/mnt/sdcard/csentry/%s", photoFilename));
```

Let’s have the interviewer take a photo of the household being interviewed in section A. Rather than adding a **userbar** button which would be easy for the interviewer to forget, we will make it part of the questionnaire by adding a dummy variable to the dictionary. In the postproc of the variable we will call *execsystem()* to take the photo. We can add a refused code in case the household doesn’t want a picture taken.

```
PROC PHOTO

if PHOTO = 1 then
    // Take photo
    string photoFilename = makertext("photo%d%02d%03d%d%03d.jpg",
        PROVINCE, DISTRICT, ENUMERATION_AREA,
        AREA_TYPE, HOUSEHOLD_NUMBER);
    execsystem(makertext("camera:%s", photoFilename));
elseif PHOTO = 9 then
    // refused, move to next field with no photo
endif;
```

This works the first time through the questionnaire but if we go back through the photo question a second time then we are forced to take another photo. We can avoid this by adding another option to the value set “Keep photo” that the interviewer will select once they are happy with the photo that they

took. When “keep photo” is selected we won’t launch the camera. We can add another option “View photo” to let them look at the photo that was taken.

```
PROC PHOTO
string photoFilename = maketext("photo%d%02d%03d%d%03d.jpg",
                               PROVINCE, DISTRICT, ENUMERATION_AREA,
                               AREA_TYPE, HOUSEHOLD_NUMBER);

if PHOTO = 1 then
    // Take/retake photo
    execsystem(maketext("camera:%s", photoFilename));

    // reenter so that interview can retake if it is not good.
    reenter;
elseif PHOTO = 2 then
    // view existing photo
    execsystem(maketext("view:%s", photoFilename));

    // reenter so that interview can retake if it is not good.
    reenter;
elseif PHOTO = 3 then
    // Keep photo - move to next field
elseif PHOTO = 9 then
    // No photo (refused)
    // Delete photo if it exists
    filedelete(photoFilename);
endif;
```

This works except that it is possible for the interviewer to select “Keep photo” even when there is no existing photo and to attempt to view a photo that doesn’t exist. We can solve this by only showing the “Keep photo” and “View photo” options in the value set if the photo exists already.

```
PROC PHOTO
onfocus
string photoFilename = maketext("photo%d%02d%03d%d%03d.jpg",
                               PROVINCE, DISTRICT, ENUMERATION_AREA,
                               AREA_TYPE, HOUSEHOLD_NUMBER);

// Only show the view and keep options in value set
// if the photo exists.
if fileexist(photoFilename) then
    setvalueset(PHOTO, PHOTO_VS_PHOTO_TAKEN);
else
    setvalueset(PHOTO, PHOTO_VS_NO_PHOTO);
endif;
```

## Viewing GPS Points

On Android you can use *execsystem* to display a GPS point on a map using “gps:” followed by the latitude and longitude. We can add a userbar button to display the household location on a map:

```
// Display household GPS point on map
function showHouseholdOnMap()
    execsystem(maketext("gps:%f,%f", LATITUDE, LONGITUDE));
end;
```

## Capturing GPS Points

To capture a GPS point first you need to start up the GPS hardware:

```
gps(open);
```

Then you request a GPS reading giving it a timeout in seconds and optionally a desired accuracy in meters:

```
if gps(read, 60, 10) then // Read up to 60 seconds, try for 10m accuracy
    errmsg("Latitude is %f, longitude is %f", gps(latitude), gps(longitude));
else
    errmsg("GPS signal could not be acquired");
endif;
```

Finally, you close the GPS:

```
gps(close);
```

In addition to querying the GPS latitude and longitude you can also query the accuracy, number of satellites, and altitude (although on Android altitude is not very accurate). See the help on GPS for details.

The longitude and latitude reported are in degrees in the WGS84 datum and are returned as decimal numbers.

Let's add a user button to update the GPS coordinates for the household:

```
// Capture current household location using GPS
function getGPS()
  gps(open);
  if gps(read, 60) then // Read for up to 60 seconds
    LATITUDE = gps(latitude);
    LONGITUDE = gps(longitude);
  else
    errmsg("GPS signal could not be acquired");
  endif;
  gps(close);
end;
```

If we want to be fancy, we can display the point on the map first so that the interviewer can view it and then decide whether or not to use it.

```
// Capture current household location using GPS
function getGPS()
  gps(open);
  if gps(read, 60) then // Read for up to 60 seconds
    // Show map so that interview can see result.
    execsystem(maketext("gps:%f,%f",
      gps(latitude),gps(longitude)));
    if accept("Save this result", "Yes", "No") = 1 then
      LATITUDE = gps(latitude);
      LONGITUDE = gps(longitude);
    endif;
  else
    errmsg("GPS signal could not be acquired");
  endif;
  gps(close);
end;
```

### Group Exercise

Build a CSPro application for the household listing questionnaire. Place your application in the folder *Popstan2020/Listing*. Treat each household as a separate case. In other words, do **not** use a repeating record and roster to try to match the paper questionnaire. For each household, automatically record the GPS coordinates in the latitude field. Give the interviewer the option of taking a photo of the household but allow for refusal. Name the photo based on the case id-items and put the photos in the directory *Popstan2020/data/listing/photos*. Go outside and test your listing application on a tablet.

## Writing and Viewing Reports

With **execsystem()** on Android you can view web pages using “browse:” followed by the address:

```
execsystem("browse:https://www.census.gov");
```

We can use this to view web pages (HTML documents) and text documents that we write out from CSPPro. To write out files from a CSEntry application we have to first declare a file variable in the PROC GLOBAL:

```
PROC GLOBAL  
file tempFile;
```

Then we can use the commands **setfile()**, **filewrite()** and **fileclose()** to open, write to and close the file. Let’s create a userbar button to write a simple text report that lists the household members:

```
// Write out and display household summary report  
function showHouseholdReport()  
    string reportFilename = maketext("%sreport.txt", pathname(Application));  
    setfile(tempFile, reportFilename);  
    filewrite(tempFile, "Household Summary Report");  
    filewrite(tempFile, "-----");  
    filewrite(tempFile, "");  
    filewrite(tempFile, "Province %d District %d EA %d Area type %d Household  
Number %d",  
              visualvalue(PROVINCE),  
              visualvalue(DISTRICT),  
              visualvalue(ENUMERATION_AREA),  
              visualvalue(AREA_TYPE),  
              visualvalue(HOUSEHOLD_NUMBER));  
    filewrite(tempFile, "");  
    filewrite(tempFile, "Household Members:");  
    filewrite(tempFile, "");  
    numeric i;  
    filewrite(tempFile,  
"Name           Sex    Age  Relationship");  
    filewrite(tempFile, "----          ---  -----  
----");  
    do i = 1 while idx < totocc(PERSON_ROSTER)  
        filewrite(tempFile, "%s %-6s %3d %s",  
                  NAME(i),  
                  getlabel(SEX, visualvalue(SEX(idx))),  
                  visualvalue(AGE(idx)),  
                  getlabel(RELATIONSHIP_VS1,  
                  visualvalue(RELATIONSHIP(idx))));  
    enddo;  
    close(tempFile);  
    if getos() in 20:29 then  
        // Android - use "view:"  
        execsystem(maketext("view:%s", reportFilename));  
    else  
        // Windows - use "explorer.exe <filename>"  
        execsystem(maketext("explorer.exe %s", reportFilename));  
    end
```



```
endif;  
end;
```

On Android we can't use **execsystem** with "view:" so instead we launch Windows Explorer (explorer.exe) with the name of the file and that will open the file in notepad for a text file and in the web browser for an HTML file.

If you are familiar with HTML you can write out HTML and have nicer formatting and pictures:

```
// Write out and display household summary report using HTML  
function showHouseholdReportHTML()  
    string reportFilename = maketext("%sreport.html", pathname(Application));  
    setfile(tempFile, reportFilename);  
    filewrite(tempFile, "<!DOCTYPE html>");  
    filewrite(tempFile, "<html><head>");  
    filewrite(tempFile, "<style type='text/css'>");  
    filewrite(tempFile, "table, th, td {border: 1px solid black;border-  
        collapse: collapse;padding: 8px}");  
    filewrite(tempFile, "</style>");  
    filewrite(tempFile, "<title>Household Summary Report</title>");  
    filewrite(tempFile, "</head>");  
    filewrite(tempFile, "<body>");  
  
    filewrite(tempFile, "<h2>Household Summary Report</h2>");  
    filewrite(tempFile, "<p>Province: %d</p>",  
        visualvalue(PROVINCE));  
    filewrite(tempFile, "<p>District: %d</p>",  
        visualvalue(DISTRICT));  
    filewrite(tempFile, "<p>EA: %d</p>",  
        visualvalue(ENUMERATION_AREA));  
    filewrite(tempFile, "<p>Area type: %d</p>",  
        visualvalue(AREA_TYPE));  
    filewrite(tempFile, "<p>Household Number %d</p>",  
        visualvalue(HOUSEHOLD_NUMBER));  
    filewrite(tempFile, '',  
        visualvalue(PROVINCE), visualvalue(DISTRICT),  
        visualvalue(ENUMERATION_AREA),  
        visualvalue(AREA_TYPE),  
        visualvalue(HOUSEHOLD_NUMBER));  
    filewrite(tempFile, "<p>Household Members:</p>");  
    numeric i;  
    filewrite(tempFile, "<table>");  
    filewrite(tempFile, "<tr><th>Name</th><th>Sex</th><th>Age</th><th>Relationship</th></tr>");  
    do i = 1 while i < totocc(PERSON_ROSTER)  
        filewrite(tempFile, "<tr>");  
        filewrite(tempFile, "<td>%s</td>", NAME(i));  
        filewrite(tempFile, "<td>%s</td>",  
            getlabel(SEX, visualvalue(SEX(i))));  
        filewrite(tempFile, "<td>%d</td>", visualvalue(AGE(i)));  
        filewrite(tempFile, "<td>%s</td>",  
            getlabel(RELATIONSHIP_VS1,  
                visualvalue(RELATIONSHIP(i))));  
        filewrite(tempFile, "</tr>");  
    enddo;  
    filewrite(tempFile, "</table>");  
    filewrite(tempFile, "</body></html>");  
    close(tempFile);  
    if getos() in 20:29 then
```

```
    // Android - use "view:"
    execsystem(maketext("view:%s", reportFilename));
else
    // Windows - use "explorer.exe <filename>"
    execsystem(maketext("%sexplorer.exe %s",
                        pathname(Windows),
                        reportFilename));
endif;
end;
```

## Exercises

1. Add a button to the userbar that opens your website (the website for your organization). Test this on Android. Bonus if you can get it to work on Windows too.
2. Add a button to the userbar that shows the interviewer manual. You can use the PDF interviewer manual that is on the course website (it only has a cover page but is enough for this exercise).
3. Add the following additional information to the summary report. Add it to the text version unless you are already an HTML expert in which case add it to the HTML version:
  - a. Total number of household members, total males, total females.
  - b. List of household assets from section G with quantity. Do not list assets where the quantity is zero.