Project DRIVE DRC

Digital Tools for
Monitoring, Evaluation, Accountability & Learning
and Project Operations

User Guide

Humanitarian Response Department (HDR)
Information and Communications Technologies for Emergencies - ICT4E
Democratic Republic of Congo, July 2014

Table of Contents

Introduction

This guide explains in detail how to use/modify all the tools made for project DRIVE DRC in July 2014. This document was done in the spirit of sustainability to ensure DRIVE DRC has a reference manual and understands important (even if small) details, hence the long size.

At the same time this document is a step-by-step learning guide, therefore I recommend quickly skimming through all sections in order at first, to grasp the concepts in each section.

Finally, all files referenced can be found in DRC's MEAL SharePoint page at **http://tinyurl.com/drivedrc** or in Dropbox at **http://tinyurl.com/ict4drive**.

The first thing to do, accountability and eValuate

Two eValuate forms should be filled out before doing anything else. Those forms are the (1) **Project Registration Form** [1] (Inscription d'un projet) and (2) the **Services Provider/Partner Registration Form** [2] (Inscription d'un prestataire de services/partenaire). These two original eValuate forms are the foundation of the CRS eValuate system. The forms improve institutional memory and accountability within CRS.

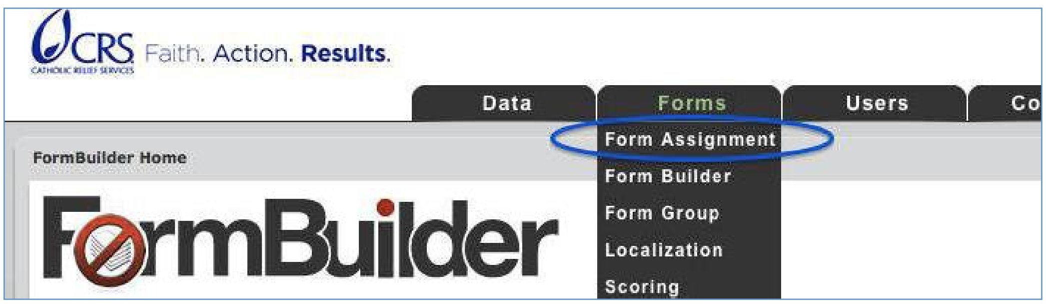
[1] **Table name:** *project_registration* – plus children. Origin: CRS eValuate – original 100% unmodified. One Potential improvement for the future: *Change the DSPN field to a SSN widget with the right mask i.e. ####.###.####.* All other eValuate forms in this guide have been modified in some way. However, an original version of all modified eValuate forms is also present within the profile. When modifying any form, I recommend backing up a copy of the original form. NOTE: when **table names change**, some functionality of the form (or other forms) may break and will have to be fixed. For more on this topic, see the section on modifying forms.

[2] **Table name:** *service_provider_partner_registration* – plus children. Origin: CRS eValuate – original plus 1 modification: (1) Automatically generate the text sting required to follow barcode QR guidelines from GKIM.

Both forms are easy to use. The project registration form is only done once per project. The other form should be filled out to register "Caritas" – the partner organization – but also to register important service providers that DRIVE will use. E.g. if a company in Nairobi will print the vouchers with holograms, that company should be registered as a service provider.

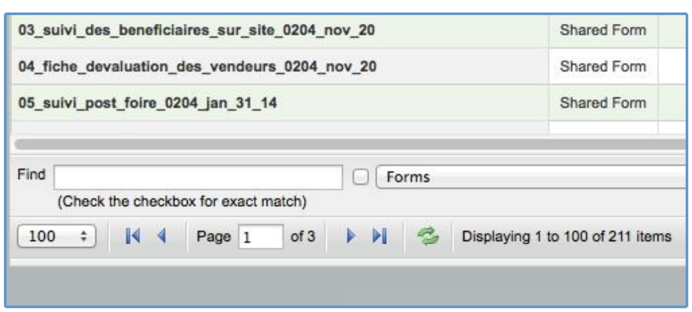
How to get these forms?

To get forms on a device, they have to be **assigned** to a username or a user group by an iFormbuilder administrator. Once within a profile with admin rights, go to the "Forms" tab and select "Form Assignment."



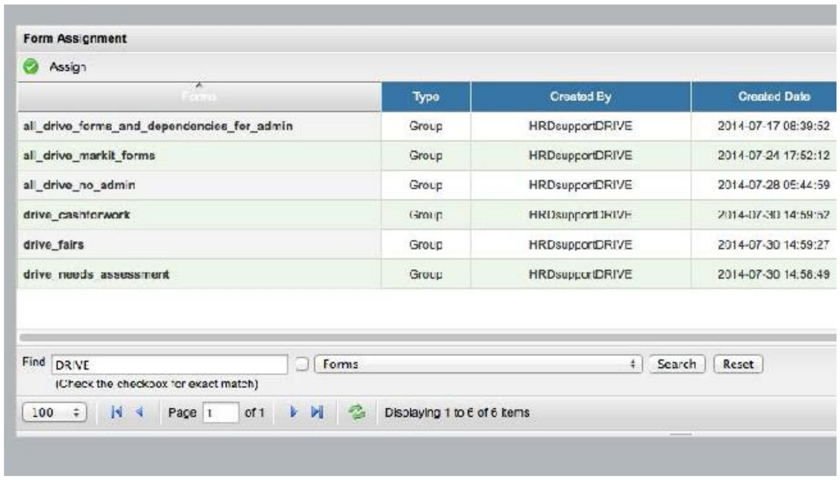
Currently, the DCR profile has 211 forms, so we have to search something to narrow down to what we want.

- *Within the "Find" search field, type in "**DRIVE.**"*[3]



[3] Typing **DRIVE** will not give you a list of all drive forms and sub forms, if you want to see a list of all DRIVE forms, type ***0214.*** *As of July 27, 2014, there are 53*

Now your selection is reduced to six choices in the following order:[4] (1) All DRIVE forms for **administrators**, (2) all MARKit forms, (3) all DRIVE forms (without the administrator forms), (4) Cash-For-Work forms, (5) the fair forms, and (6) the "needs assessment" with the French outputs. These choices are pre-programed form groups, they can be modified if you like.[5]



Supposing that you are the administrator, click on the first[6] option and click the "Assign" button. Now you have to select a user, and there are multiple.[7] But fortunately some of them are named with the key word DRIVE. But you can assign a from group to any account.

For example: (1) If you need to assign all DRIVE and admin forms to Katherine Price, you assign the group to "**PQMGoma**" (both the **collect** and **view** rights – see screenshot). If you need to assign the same forms to Giulia Frontini, you assign the group to **"elmict4d."**

[4] The actual names that you will see can be seen in the screenshot. IFormbuilder uses a database friendly machine name to save all form groups.

[5] One example of a change could be the needs assessment group. E.g. change group to just show the English output version of the needs assessment. These changes are made in the "Form Group" option in the "Forms" tab

[6] all_drive_forms_and_dependencies_for_admin

[7] As of July 27, 2014, there are 98 users

The DRIVE usernames

- There are 44 usernames with the keyword DRIVE. 40 of those usernames are DRIVE plus a number, i.e. **DRIVE1, DRIVE2** … **DRIVE40**. In addition to those 40 accounts, there is **DRIVEdatabase** (for the database officer), DRIVEcashier,[8] DRIVEmeal-goma, and DRIVEmeal-pweto. The last two were made for two MEAL officers.
- There are also two[9] drive groups, **drive1to20** and **drive21to40.** If you assign a form or form group like "Needs Assessment" to the first one of these, 20 of the iPads with the accounts DRIVE1-DRIVE20 will have the Needs Assessment assigned to them.

Back to the first thing we should do

After we assign all admin forms to **PQMGoma** or another account for a manager, 12 forms will download in their devices including the (1) **Project Registration Form** and the **Services Provider/Partner Registration Form**.

The right settings for each iDevice

Almost every component of the DRIVE forms was designed for a (100%) offline environment, i.e. without the need for a thunderplug mini-server. But many forms are fed data from databases, not barcodes,[10] so make sure you sync all forms at least the day before you use them, but preferably before that. Ideally, you want to use the best connection available, so you might need to sync days before the "big day," if there is travel involved.

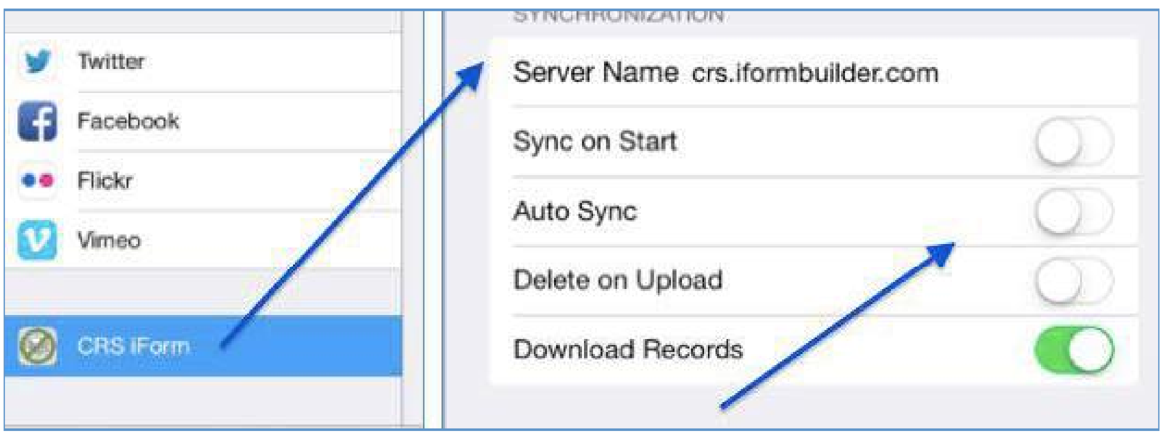
[8] To be explained at the finance table (if paying at the fair) section

[9] If you count the drive_forms_review group, there are three. This group will also assign the forms to Kathryn Clark and various members of GKIM across the world.

[10] These forms do scan barcodes, but the barcodes are only coded with the **household ID number,** not all their data. This was done for one main reason: **the simpler the data, the faster the bar code reads.**

You may set all the iPads to airplane mode, which saves on battery life, however this shuts down the GPS. **Alternatively, you can manually shut down all wireless radios (Wi-Fi, Bluetooth, Cellular data) and the location widgets still work.** The fair forms for example, while they don't have location widgets, *they collect the coordinates of the place where the form was first opened and modified.* This data might be of interest when reporting after the fairs.

Most if not iPads should maintain these settings in the field, **with the smallest picture size**.



Why use the mini-server

1. The payment forms (for the fairs) are automatically generated. If you want to pay vendors on the spot, it would make things easier. It enables a separation of tasks, where the person that first counts the vouchers is not in the same location as the cashier – *while leaving a paper trail.* The iPads would then wirelessly send the form

almost completed to the mini server. Then, in a laptop[11] with a small portable printer[12] one could download the PDF, print it, and "sign the original documents." One also could copy the information calculated into a paper form, verify that it is correct, and thus not use the mini-server. If there are be less voucher payments than beneficiary admissions[13], this might not be a bad alternative after all.[14]

2. The second reason to use the mini-server would be to download data **in the field to make and distribute ID cards on the spot.** This requires a laptop, the Bartender application with a license and **a printer**. Same as with the payment form, you download the excel file that you can use with Bartender and a premade ID template.[15]

The iPad settings for the mini server change a bit. (1) Make sure the mini-server and the iPads were synced with cloud at the same time (the information should be the same in both). The database officer was trained on all this,[16] but the key settings and steps are the following:

1. **Close the CRS iForm application (and all applications) from the device before changing the settings.** Double tap (quickly) on the **home button** to see the task manager. Flip the windows **UP** to close them. Close all applications.

---

[11] **This laptop could have an iFormbuilder account that only assigned to "View" the payment form table, making things even easier for the user.**

[12] I would be interested in trying the **HP Officejet 100 Mobile Printer**, unless I could find enough renewable power to run a laser printer.

[13] Almost always the case

[14] Vendors could also get paid at a later date, making things even LESS tech or work hungry.

[15] More details in this topic…

[16] https://iformbuilder.zendesk.com/hc/en-us/articles/201702810-ThunderPlug-Setting-up-the-Plug-D3-

2. In the iPads **"Settings"** app, go back to the CRS iForm section and use the following server name: **192.168.40.100** (Close the settings and double-check to make sure the settings were saved)

[17]

3. Join the WiFi Network of the *mini-server*[18]

4. Sync "CRS iForm" once with the mini-server to test *the mini-server* before going to the field, to make sure it works.

[17] Auto Sync: It automatically syncs the iPad with the mini-server or the cloud after a "**parent form**" has been filled out. However in the time it syncs, the device cannot collect data, which can be a loss of precious minutes per person. Test it out and see what you think – I think it's best to keep it OFF.

[18] The mini-server has two modes. Offline and Online. If you plug in the mini-server with the Internet via cable, computers can use the network for the Internet as well. If the mini-server does not see the Internet when you first turn it on, it will not try to connect to the Internet. If you want to use the Internet (while in offile mode), you have to unplug and re-plug. In the field, the device operates in offline mode.
SSID = *CRS-D3Plug-38-cc*
Password= *iformD3plug*

Simulating and Testing, making sure the devices have the right forms and they have an account.

Do a final test on all forms before sending them off to the field. But before that, you want to do simulations with the users. Collect fake data and practice. Then you erase all the practice fake records from the cloud.[19] *Afterwards, erase the local database* (in the iDevice – look at screenshot). **All this should only be done before using any form for the first time in the field, and then the table should be kept untouched for life.**[20] NOTE: DO NOT ERASE DATA FROM THE BENEFICIARY AND VENDOR DATABASES before the fair (unless the data is fake).

Once you re-sync to load all the forms in each device, you want to quickly go over all the forms in the iPads before they are set free to the field. Test scanning barcodes, but don't save records in the forms. This is very important, since there is a possibility that an iForm client had trouble downloading forms if the Internet was intermittent, making the app believe incomplete forms are complete – when the forms are actually missing major pieces. If this happens, you should "erase the local database," and sync again until the forms look and act right.[21]

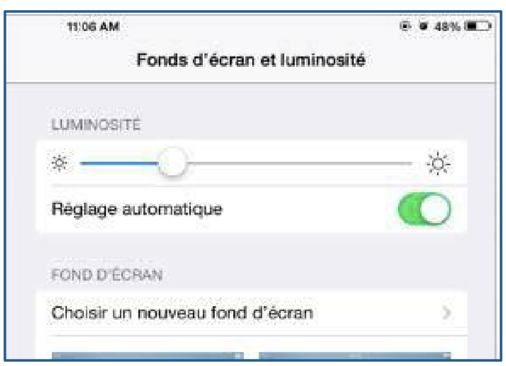
[19] See second screenshot in page 27

[20] Once a form is live and in the wild, it is best to avoid all modifications possible. Changes to elements can erase all data for that column. If you want to modify a live form, make a backup copy. You may want to backup all your records too.

[21] All forms should be tested and re-tested. The output files should also be looked at. However before going live for the first time, all form tables e.g. fair or cfw tables should be emptied to **erase all the test records.** I say this assuming all the iPads already were checked to make sure they are linked up to accounts.

Battery life saving tips that also improves performance.

1) **Shut off all wireless radios that are not needed**. It's hard to know what you need, but you can do almost everything with DRIVE forms even if you shut down everything. You have the following options depending on what's available at the site:
   - Airplane mode (Best battery life)
   - **All wireless radios OFF, but no airplane mode (GPS coordinates get captured)**
   - Wi-Fi ON and all other radios OFF **(The mini-server communicates via Wi-Fi)**
   - Cellular Data ON, all other radios OFF (If the 3G connections is like Lubumbashi's 3G Orange network, it would be a very good idea. This option would be better than the mini-server, but highly unlikely).
2) **Close all applications, except for "CRS iForm."** If the application does not work well for some reason, save your work, and quit the application – then re-open it. The more open apps, the faster the battery and processing power gets used up.
3) **Keep brightness to the minimum level that's confortable.** If you are under the sun, you will be forced to use the highest level.[22] **Try to use the devices under shade**.



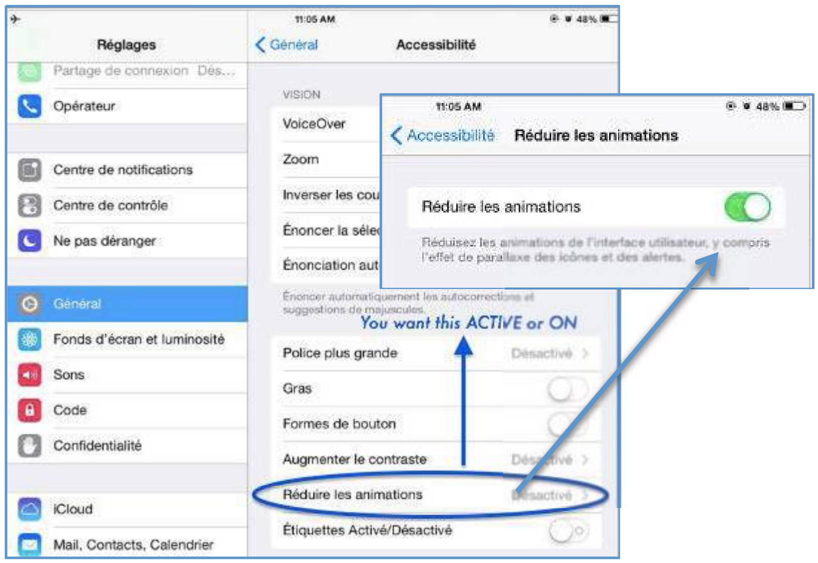
23

24

[22] If you stay under the sun without shade for a long time, the machine will overheat and inform you about it. To prevent hardware damage, the iPad will shut itself off and you won't be able to use it until it cools down. **The best thing is to have shade whenever possible – and plan to make it possible.**

[23] Brightness settings within the iPad's main "Settings" (Reglages) app

4) The iPad's operating system – iOS7 – has cool animations that consume more resources. Disabling them will help improve battery life. In the "General" section of the "Settings" app, go to "Accessibility" and you will see the option to reduce animations.



5) If you are using the Cellular Network, shut off the 4G LTE radios. LTE does not (yet) exist in the Congo and the LTE radio consumes more.

[24] Quick access to brightness by flicking your finger from the bottom of the screen UP, but this is hard to accomplish with the Otter and Survivor cases.

Troubleshooting strategies

1) **For both the iPad or Mini-server, shutting down and restarting can fix most if not all issues.** To force the iPad to reset, *hold both the "top" power button and the home button for 10 seconds.* **Make sure all field agents practice forcing a reset.** With the mini-server, one just *unplugs and re-plugs.*

2) If the CRS iForm application keeps crashing, close the all applications and restart the iPad.

3) If the CRS iForm hangs when opening, make sure all your wireless radios are OFF and you iForm settings are correct. The application hangs at opening, as it tries to "sync on start" when the connection is poor.

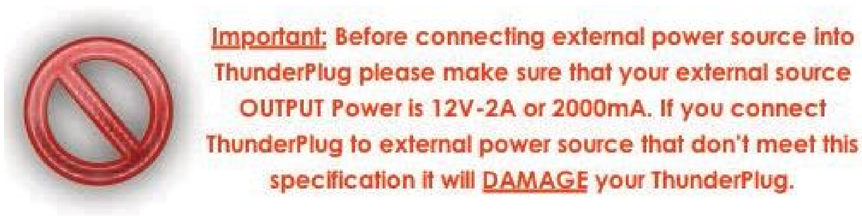
4) If forms did not download correctly, "Erase local database and re-download.[25]"

5) If the iPads are slow after collecting data for days, sending the data to the mini-server or cloud and erasing the local database will speed up things up.

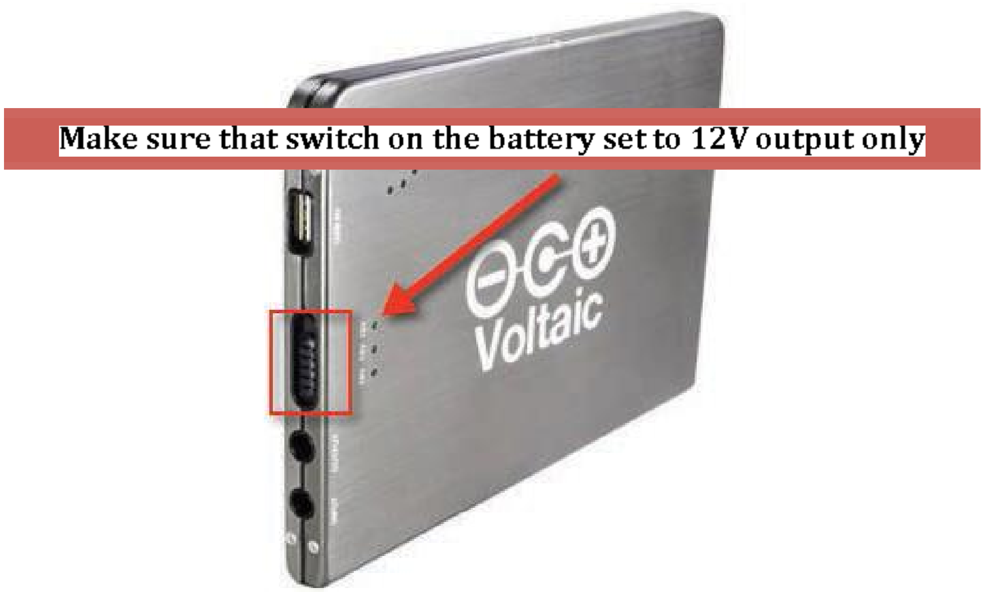
[25] One can also sync the forms from the mini-server.

Mini-Server with External Battery (Voltaic Solar Panel, battery, and mini-server come separate)

The mini-Server can run with the voltaic solar panel and batteries.



1. Insert one end of the cable into battery into "OUTPUT" socket on the battery



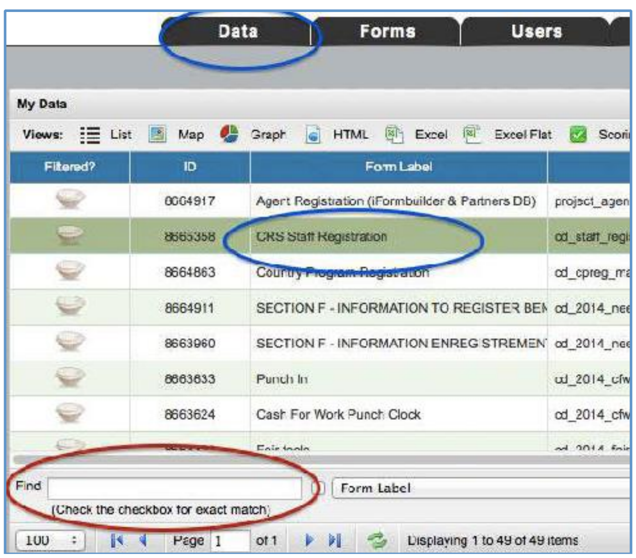
2. Connect other end of the cable into Thunder Plug power socket

Register CRS Staff

The second step in the process is to have all CRS employees that will be working with DRIVE registered with the "CRS Staff Registration" form. This form is also very straight forward, but it can improve in the future.[26] The icon is a person with the CRS logo and a plus (+) sign. The form has fields where it asks to scan the hiring officer's number and supervisor's number. Since this form is brand new, the first few times you might need to "type" the officer's number, until those other staff members are also registered.

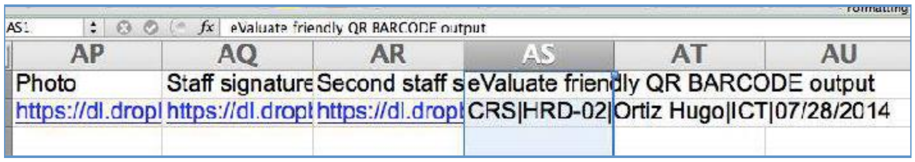
Once the form is filled out, the excel file has a column with the text needed for to make a barcode. Look at the instructions and screenshots below to understand this detail.

1) Log in with an account that can view the CRS Staff Registration table, in our example it's **PQMGoma.** Click on "CRS Staff Registration" within the "Data" tab of CRS iFormbuilder. In the event that you don't see the form, you can search for it with the "Find" field by looking for a key word from its label name: **CRS Staff Registration** or its "table name."



[26] This form was the last request from the country program; currently the Staff ID number is not automatically assigned. This was left as a text field to allow an existing CRS staff ID number. Also some option lists were left in English like the job titles, some (not all) of these should be translated into French.

2) Select the CRS Staff Registration from the list and click on the excel button to download its data as a spreadsheet. Then if you want, we may look at the excel file to see what comes out. *The last column is for an automatically generated string of text, which is very similar to the official CRS GKIM QR structure guidelines.*[27]



How to make a barcode

There are many ways to make a QR barcode,[28] but given the volume of barcodes that CRS makes, the official way is the best (with Bartender). Even if your computer will not have a license for the software, it's worthwhile to download the trial version, which allows you to make and view barcodes, but not print them.

1) Open your preferred browser and download bartender at:

http://www.seagullscientific.com/label-software/barcode-label-printing-software-download.aspx?lang=en

2) In an African context, it makes more sense to download the "Full Installer," and share the installation file with all people that will potentially be making or reviewing bartender files.

3) During installation, you will be asked to choose the edition that you would like to install, i.e. the basic, professional, automated… **Choose the "trial" version with all versions included.** A trial version can be made into a paid professional version once the license serial number is purchased and acquired by the project.

---

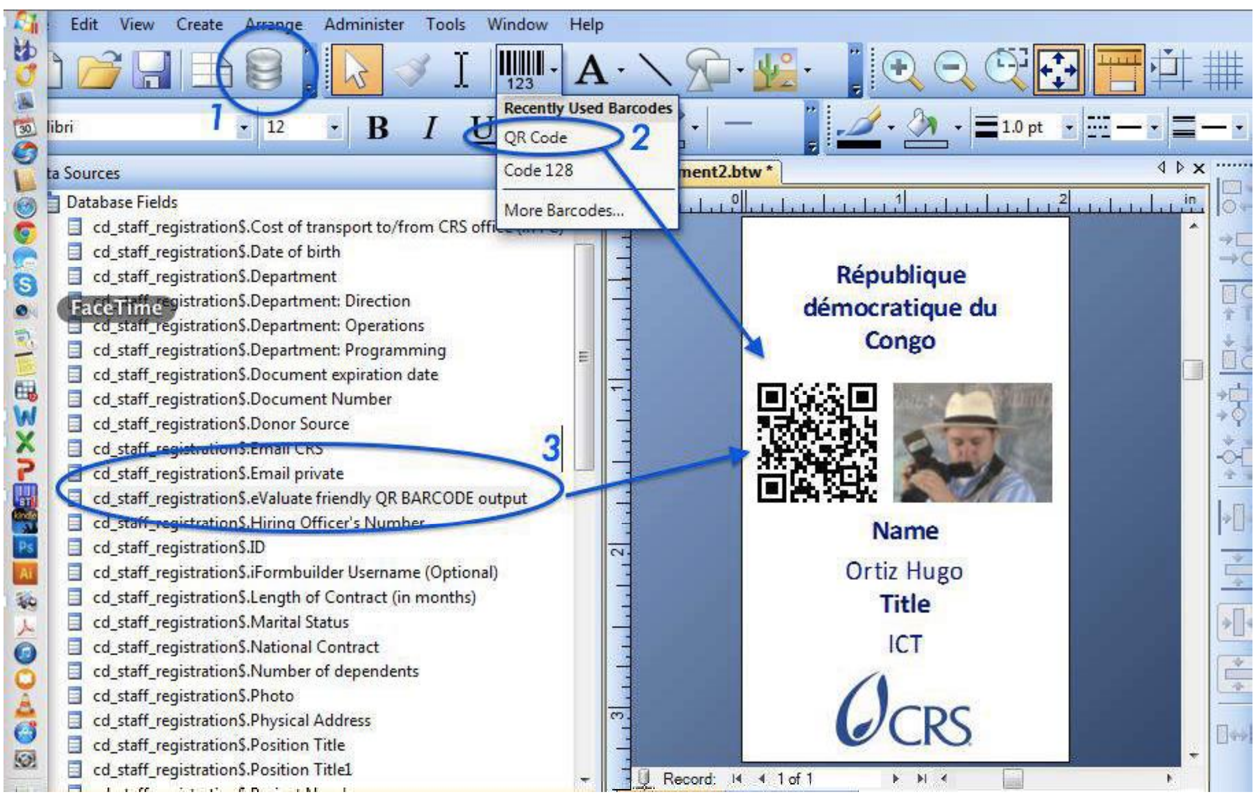
[27] https://global.crs.org/teams/GKIM/MEAL%20ICT%20Platform/Shared%20Documents/QR_barcode_standart_structure.docx

[28] The database officer for project DRIVE already knows how to make barcodes and has the right software installed in his computer. As of July 28, he was only missing the license serial number.

4) There are multiple guides to bartender within CRS global,[29] including some that were made in the DRC, thus in this guide I will not go too deep into the basics. Also, the database officer was trained on Bartender and should be able to do following the steps.

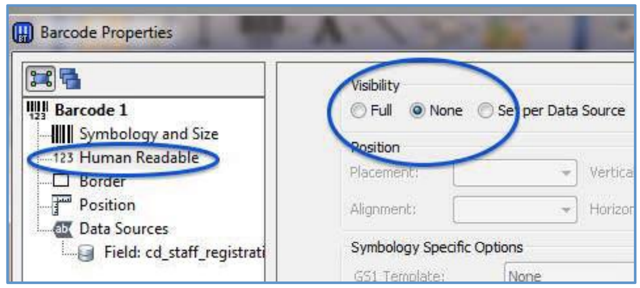
5) After you made a design for the card or opened an existing bartender document (with the design that will house the information for the staff members), make sure to add a QR barcode element (#2 on the screenshot above).

6) Then, link the bartender file to the excel document (our staff registration) "database" file that you downloaded from CRS iformbuilder. To link staff registration file, click on the database icon (#1 on the screenshot).



29 The following are links to different bartender guides, the second file is an English guide made in the DRC, the first is a different guide in French that teaches bartender with iFormbuilder practical exercises, like how to make an ID card: (1) https://global.crs.org/teams/DRCMEAL-ICT4D/MEALICT4D%20Platform/DRIVE%20ICT4D%20solution/Existing%20CRS%20Guides/Bartender%20IDs%20(FR).pdf (2)https://global.crs.org/teams/DRCMEAL-ICT4D/MEALICT4D%20Platform/3D%20Fair%20ICT4D%20Solution/CD_2014_Bartender_user-guide_FINAL.docx

7) Then “code” the QR code element (from step 5), and drag the column: “**eValuate friendly QR BARCODE output**” from your excel document into the QR code element you created. You may want to **hide the Human Readable output** by double clicking on the barcode element and disabling its visibility. If you are happy with you staff ID card, you can print it, or you can just scan the barcode from your computer screen.

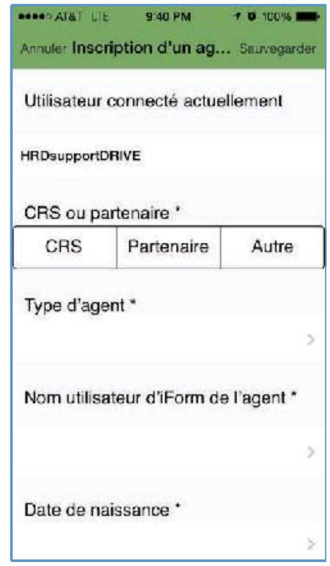


A quick way to make just one barcode

While bartender is the best way for CRS to make barcodes due to the high volume, there might be a time when you quickly just want to make one barcode, to test the system or to quickly get started. If you copy the information from the **eValuate friendly QR BARCODE output** column in you excel file, you can then go to http://www.qrstuff.com and make a barcode there.

Just select “Plain Text” as the data type, paste the information to code into a QR barcode, and the QR barcode will be on the right.

Register field agents: Assign CRS staff iFormbuilder accounts and partner field agents



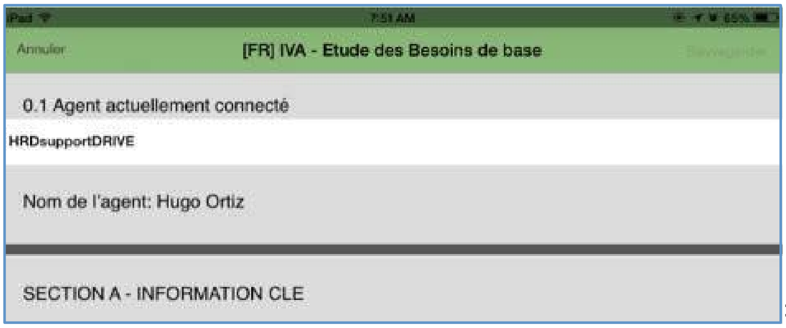
Once you filled out the **Project Registration Form,** the **Services Provider / Partner Registration Form** and/or the **CRS Staff Registration Form**, you can now register field agents. If you are registering a field agent from a partner organization like Caritas, you should follow the same instructions to make a barcode as the instructions above, but with the service provider / partner form. The **Services Provider / Partner Registration Form** was modified to automatically make a column with the right information to make an eValuate QR barcode, just like the "CRS staff from;" otherwise, this form is the eValuate original.

Once you have a partner barcode or a CRS barcode, you can start using the **Project Agent Registration Form**[30] **(Inscription d'un agent de projet).** This eValuate form[31] makes

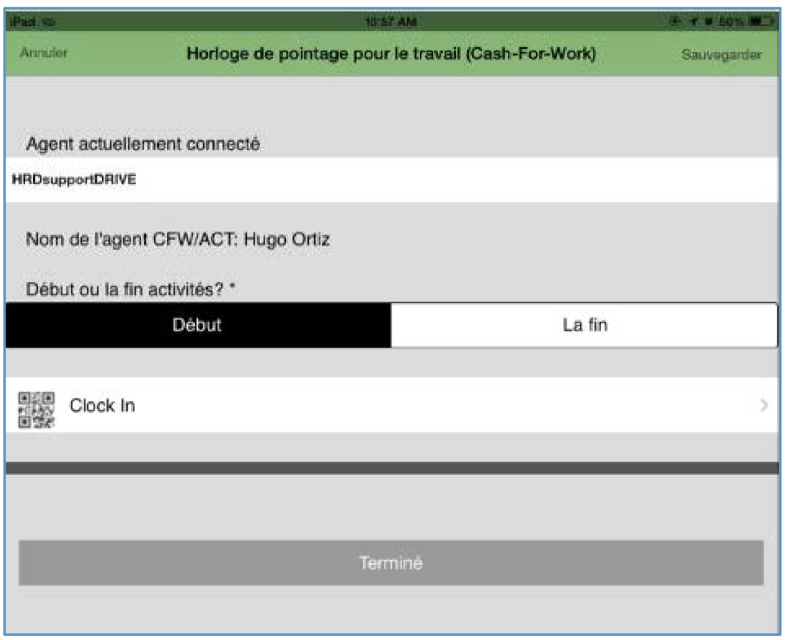
[30] In the DRC profile, this form (the modified version of the eValuate form) was renamed in English to: Agent Registration (iFormbuilder & Partners DB). The original was duplicated but retains the original label.

[31] This form was modified in five ways: (1) French translations were changed, (2) empty barcode element was hidden from output files (excel), (3) added the possibility to register field agents as CRS employees [this also hides fields of known information], (4) added the possibility to register field agents without scanning a bar-code [other], and (5) the icon was changed.

accountability of field agents – easy – once they're registered.[32] Within this form, you will assign the username and register important information about the field agent. If you scan a CRS barcode, most of information is populated automatically or not asked[33]. **Once you have done this, you can use any "eValuate form"** and **DRIVE forms.**



[34]



[35]

[32] For this to work, people always have to use their own iFormbuilder account, this usually means that agents must not share their devices. Once an agent is assigned a device, they are accountable for the data collected in that machine.

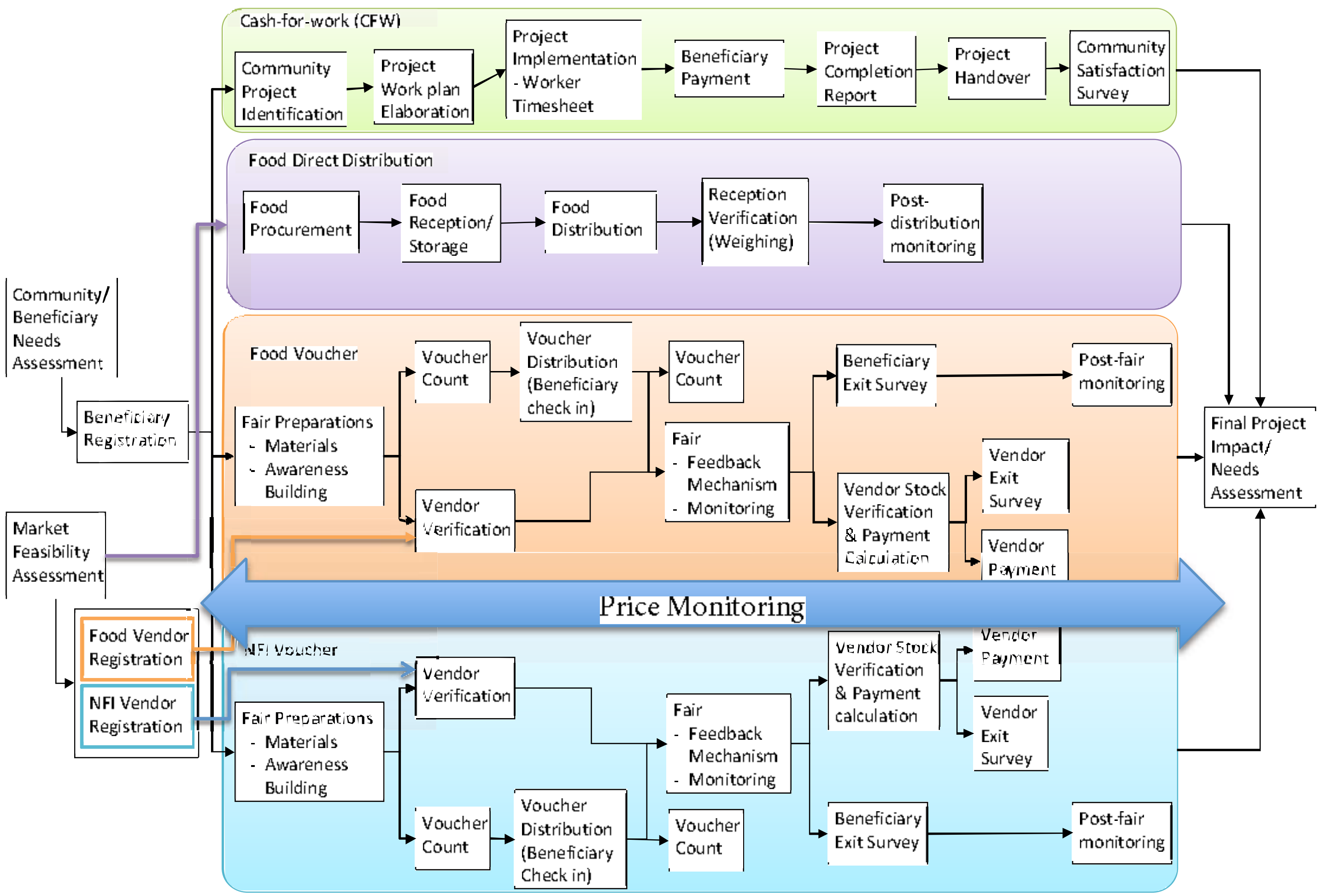
[33] Feature request from Kathryn Clark ☺

[34] **The screenshot above is from the DRIVE needs assessment form. While this is not an eValuate form, it acts like one by not letting the user complete the form unless the users are registered and accounted for.**

[35] Cash for Work form, another DRIVE form behaving like an eValuate form.

Finally!!! – Project DRIVE DRC's process map

While project DRIVE DRC is a complex emergency recovery program comprised of "non-food-items" and "food commodities" fairs, non-conditional vouchers, conditional vouchers, conditional transfers, non-conditional transfers, and possibly direct distributions; on June 4th a simplified process map was developed to showcase a break down of the project's key task.



All tools in this document starting from this point are pieces within the process map, however some of the forms needed for tasks in the process map have not yet been developed. Additionally, not all tasks within the process map require a form.

Needs Assessment

Introduction

DRIVE DRC will do a “needs assessment” in order to help identify the project’s beneficiaries and to determine the most effective and appropriate emergency response for them, within the project’s proposed guidelines. The Integrated Vulnerability Assessment (IVA) was adapted for project DRVIE DRC in July 2014 by Giulia Frontini, with the help of Katherine Price and Michee Kashoshi.

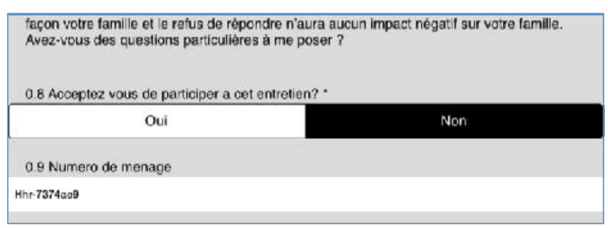
Two versions of DRIVE DRC’s needs assessment were developed in iFormbuilder:

- cd_2014_needs_assesment_master_0214 (from the screenshot below)
- cd_2014_needs_assesment_master_0214_en

Both versions should look the same for the data collection field agent. Moreover, both are in French and English, however the first one makes French output files – i.e. excel files, PDF files, online feeds – while the second makes English outputs.[36] Only the French version was included within the DRIVE form groups, since the managing team expressed the need of French outputs to share with the cluster[37] and local partners, however the English output version can be enabled by any staff member with administrative rights within the CRS DRC MEAL Profile.[38]

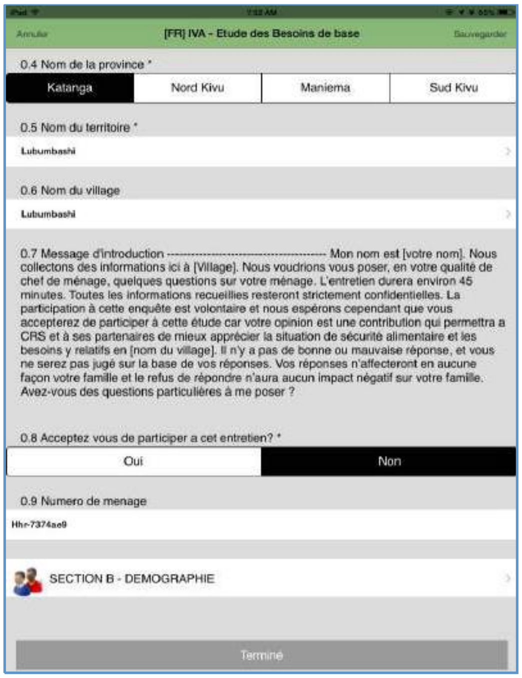
Using the form

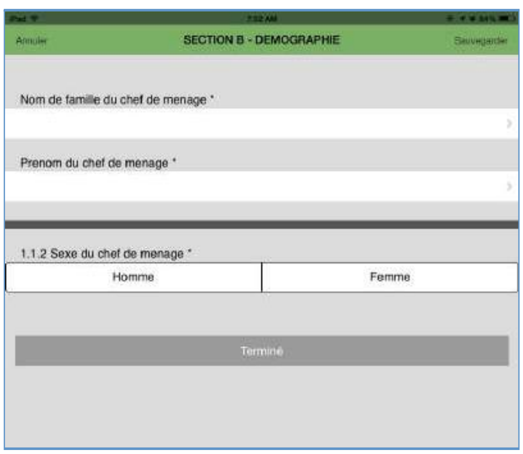
This form should be used to survey an entire population, not just a sample. However, the form is long and thus is broken up into multiple sections (sub-forms). Once the user fills in the parent form to record basic geographical information, it presents the agent with an introductory message for the interviewee. Then it gives the interviewee the opportunity to opt out of the survey and thus surrender any possible benefit from project DRIVE DRC (question 0.8, **do you accept to participate in the survey?**). If the user accepts to participate, all the sections will be appear. Otherwise, only a very small part of the demographics section will be made available to record the name and sex of the head of household. Either way, the form will assign a household number to the household.

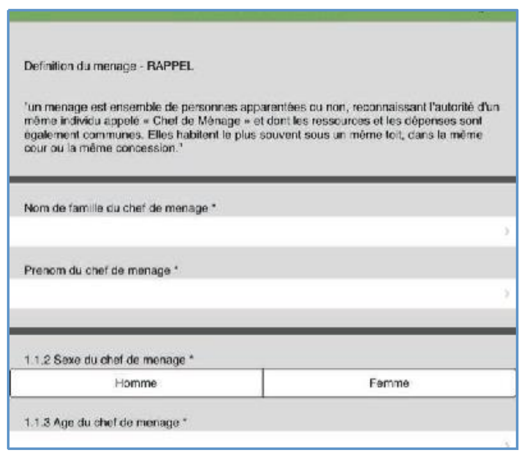


[36] There is a bug in iFormbuilder version 6.3.2.7 and earlier, which erases translations of option lists if clicked on in the option list manager. Due to loss of time from the bug and multiple translations, some option lists in the English output version were kept in French.
[37] Non-state actors intervening in country that coordinate activities and share information within a sector like non-food-items – to reduce duplication of same activities and inefficiency.
[38] The profile number is 156784

The screenshot below shows the difference for people who deny (left) and accept (right) to participate in the survey.



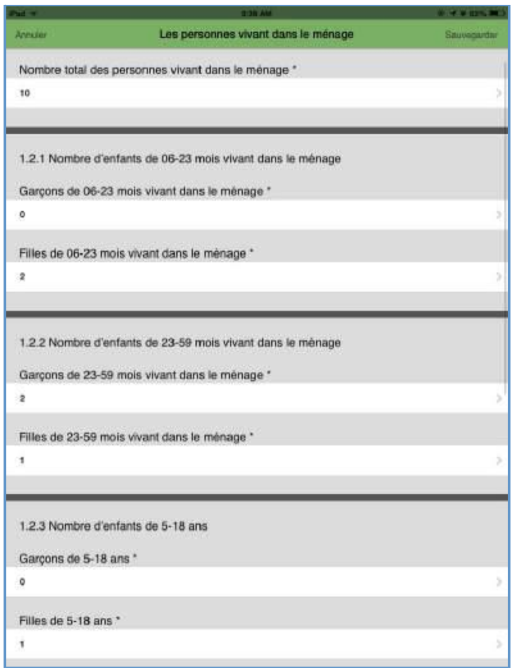


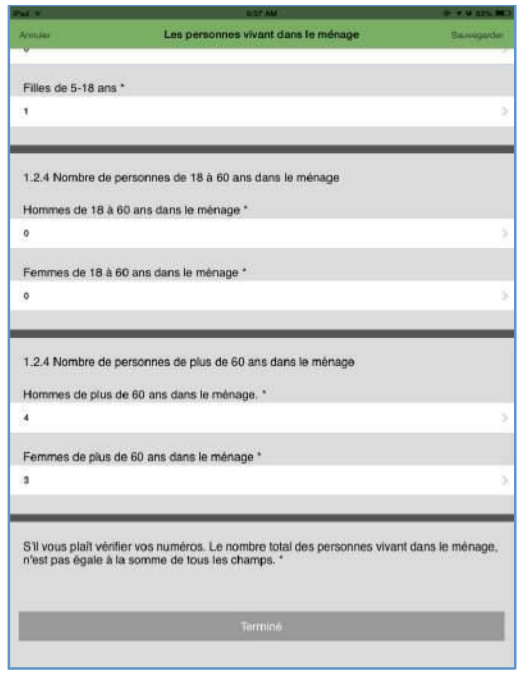


The form is easy to use (including its children sub-forms). However some areas of the form need to be explained:

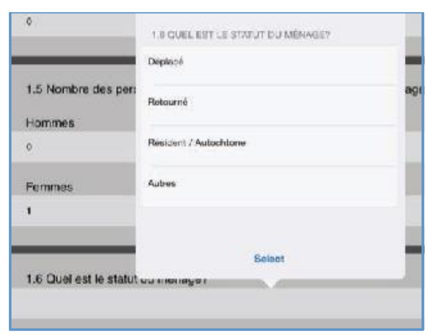
1. Within the demographic section, there is a nested sub-form (a child sub-form of the demographics sub-form). This sub-form helps accurately count the members of the household by age and sex. The first question in the sub-form asks for "the total

number of household members;" consequently the agent has to enter the number of members within different age and sex categories. If the total sum of household members within the breakdown categories, does not equal the number of total household members from in the beginning, the form will instruct the agent to re-look at the numbers to fix the mistake.

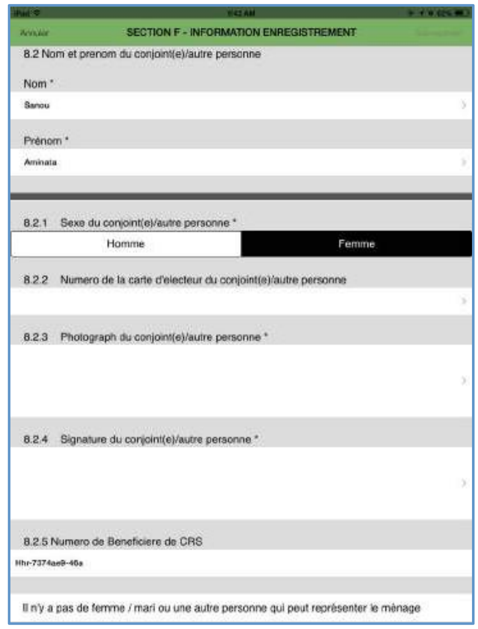




2. The question about the household status (question number 1.6), will determine most of the skip logic for the rest of the form.



3. Section C, which assesses the household's vulnerability with respect to essential household non-food items, will calculate the NFI score-card (both final and for each category) automatically. However the score-card will not be shown to the data collector. Instead, it will can be seen in the excel output files within https://crs.iformbuilder.com.



4. Towards the end of the form, one has to register personal information about the head of household and the spouse (or other person). In many situations, the spouse or other person will be the future DRIVE DRC beneficiary.

   - Once the Needs Assessment form is complete, it creates an eValuate friendly barcode for the spouse (or other person), and it makes a separate barcode output for the head of household, just in case they become the beneficiary.[39]

5. When you download the excel file, there will be multiple sheets. One for demographics, NFI, food security... The last one contains the registration information.

8.2.1 Sexe du	8.2.2 Numerc	8.2.3 Photogr	8.2.4 Signatu	Date de naiss	Conjoint ne sa	8.2.5 Numero	Il n'y a pas de	eValuate frien	eValuate frien	pa
Femme	32456789086	https://dl.drop	https://dl.drop	0/12/1990	1	HHh-399df-40	0	BEN|HHh-00§	BEN|HHh-00§	

[39] The barcode would authenticate with eValuate forms anywhere, and thus provide the Name, ID, etc. This is practical since it works even in offline environments without a database. Any CRS iPad with eValuate forms will see that barcode as beneficiary data. However in some scenarios these barcodes may be inappropriate, since just reading the Barcode can give you a breech the information. Many phones now days come preloaded with QR code scanning applications, since Quick Response went mainstream. **If data security is a priority, it is best to code barcodes with just a token ID number.**

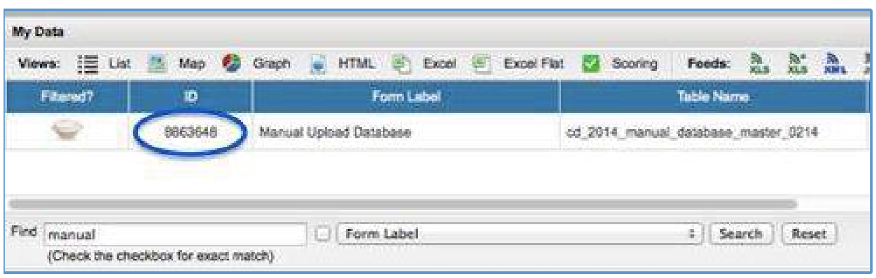
Beneficiary Registration

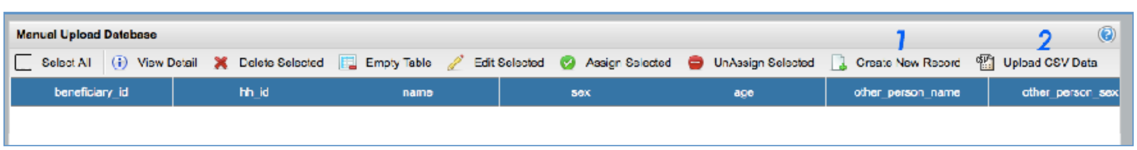
Multiple ways to register beneficiaries were built-in for DRIVE forms. In an ideal scenario, after an analysis was done with the needs assessment data, you would go back to the community to ensure local leaders and community members are involved and have a say in the beneficiary selection process.

There may also be situations where a partner gives data to you today – and tomorrow you have to act. For these two reasons, the biggest level of flexibility was put in place to this task. Either way, a beneficiary database is a requirement for almost all other forms. This is what allows the other forms to be QUICK.

The manual update table

Log on to https://crs.iformbuilder.com with an **admin** account that can view[40] data in the manual upload table. You can search on it by name. Click on its **ID number**.



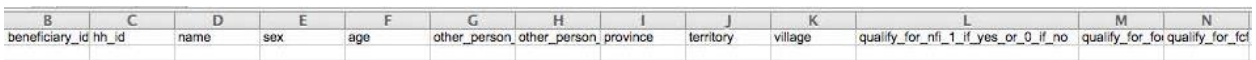


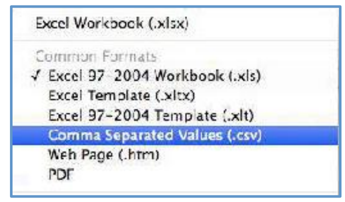
Then, click on **"Create New Record,"** then **"Save"** to create an excel template that will be uploaded as a CSV Data. Once you have created an

[40] Three accounts already have access to view the data in this table, PQMGoma, elmict4d and DRIVEdatabase.

empty record, you can go back to "My data" and download an excel file, just like you have done for the other forms and their tables.

Within the template, you can paste data from other files, just make sure the right data is in the right column, i.e. the name of the beneficiary is in the name column. Also, make sure you keep the title row with its original names.





There are columns to register if beneficiary qualifies for **X** benefit of the program, in these 1 means yes, 0 means no. Then, when you save the excel file; change the file format to "**csv**," or comma separated values. Now the file can be uploaded, by clicking on the "Upload CSV data" button (screenshot).

An alternative way: Borrowing from eValuate, the beneficiary registration form.

One of the greatest eValuate forms is the beneficiary registration form. The latest version is a quick needs assessment tool. (Example of ideal DRVIE scenario in italics) *After the analysis gave you a good selection of beneficiary candidates, the database officer is told to make booklet of potential beneficiary pictures, with their barcode, name and household size.*[41] You don't have to exaggerate like the picture on the right by fitting 32 potential beneficiaries to a page.

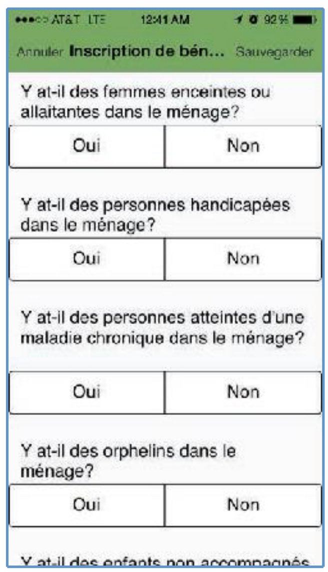
Once in the community, the picture is a tool to engage leaders and others into being involved in the beneficiary selection process. They could verify that the farmer you thought were

[41] The database officer should be able to do all this. He replicated a similar exercise multiple times. Just keep clear what information will be encoded in the barcode, in this case it is the eValuate friendly field.

The picture could also take the whole card and the barcode and name could be in the back.

*going to help is actually a government schoolteacher, **OR** that someone who was missed is actually within the criteria of most vulnerable and needs the most help.*

The DRIVE version of this form looks a lot like the eValuate counterpart, however it has the geographical locations for the Congo, it asks questions pertinent to project DRIVE, it allows to select if the beneficiary is a "skilled or unskilled" worker if they qualify for Cash-For-Work. Missing translations were added for certain questions, plus some other tweaks.[42] If you select to import from the needs assessment (IVA), *you can scan the barcoded pictures* ☺.



[42] This form still looks and feels like the original, but it underwent many modifications. Most are invisible to the user, but most importantly it feeds data to the fair forms and cash for work forms. This is also the only eValuate form that has been renamed to signify significant modifications and because it is customized specifically for project DRIVE + livelihood projects.

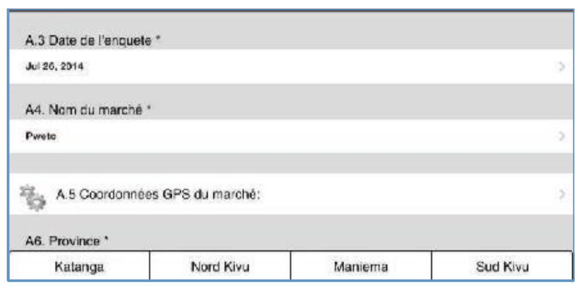
MARKit – Market Assessment Forms

Three Market Assessment tools were adapted for DRIVE and iFormbuilder, a Market registration form, a price-monitoring tool and a market study survey.

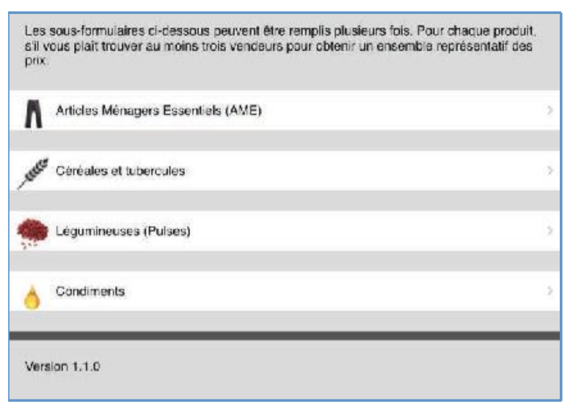
These forms are straightforward, but some details will be highlighted about them.

The market registration

- The **market registration** form collects GPS coordinates within a sub-form. Make sure the **device is not set** to Airplane mode. Also, make sure **location settings** are active for CRS iForm.[43]



Price monitoring tool



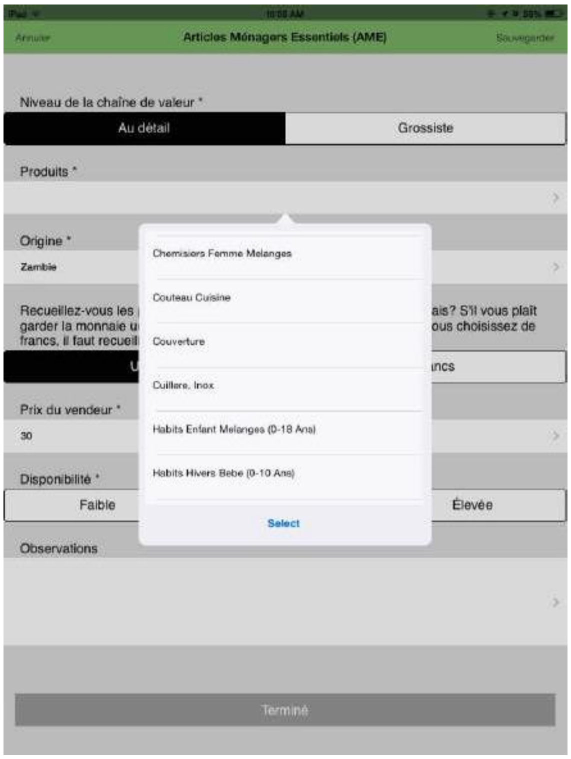
This form was built and re-built with teamwork in mind after testing another version in the Pweto Market.[44]

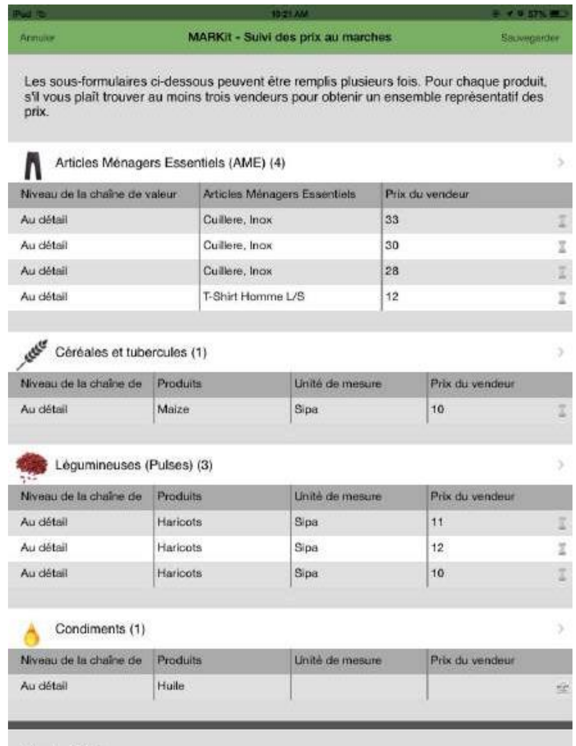
A team of people divide and conquer by

[43] The very first time an iPad opens the CRS iForm client, it will as if you want to enable location settings. Make sure this was accepted.

[44] This is by far one of the best DRIVE forms.

having different teams collect data on different types of products; one team collects on non-food-items while another does the condiments. *One person could also choose to collect everything.* **Price monitoring should happen throughout DRIVE DRC's life.**





As you can see in the screenshots above, each sub-form is very simple but changes depending on the product category. **For this to work well, all data collectors have to be instructed to find three vendors in the same value chain level for each product within the market.**[45] Meaning that if you are interviewing "pants" retailers, you should get prices from three pants retailers. But if you do a pants wholesaler, you should also get three pants wholesalers. All these sub-forms can be filled out as many times as you need them.[46]

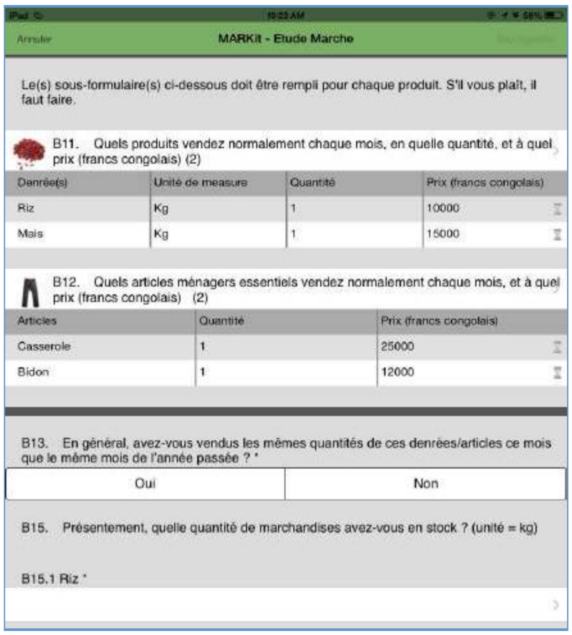
[45] Unless there aren't three vendors within that market

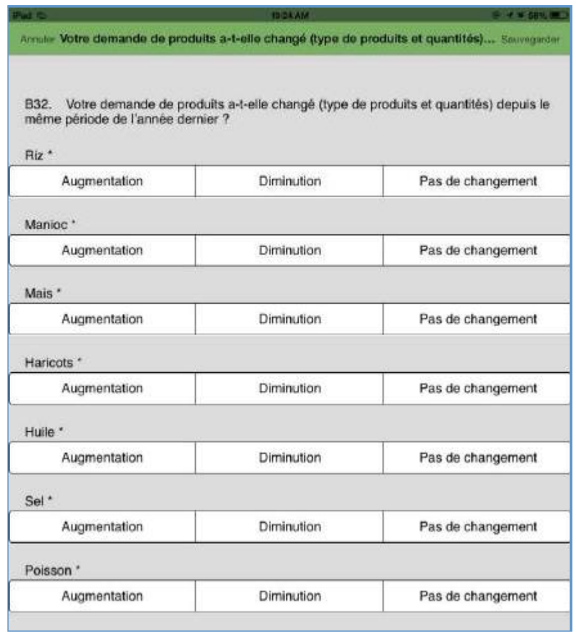
[46] To be extra safe, please instruct the data collectors to save constantly. To save your work, click on the top right button at any time

Market Assessment

This is the longest of the bunch, but it contains a lot of skip logic, depending on what you choose, it can become the smallest of the bunch.

One thing to note on this form is that some sub-forms can be and should be filled out multiple times – while others only once. Also, this form contains many select elements to speed up the form.





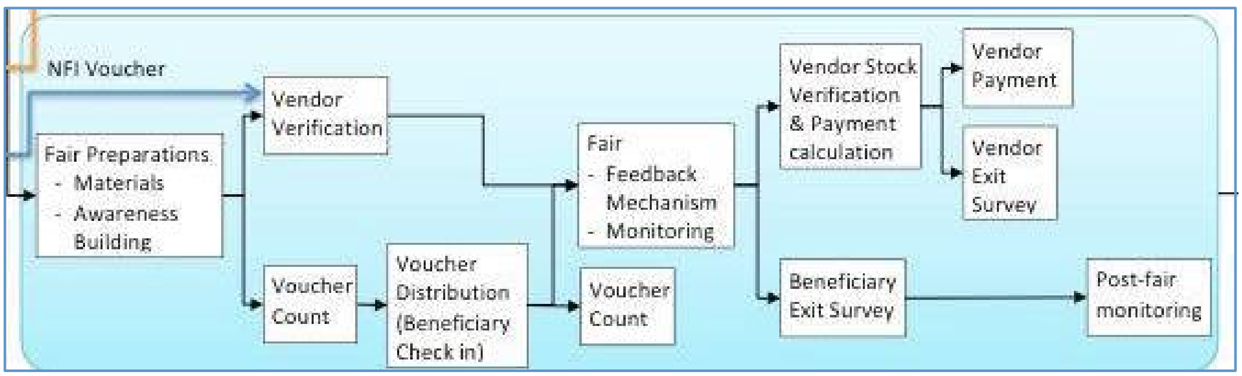
Vendor Registration

This form is very simple, but it's super important as it makes a database of all vendors in the DRIVE Fairs (or distributions) forms. It feeds the right information and skip logic to all DRIVE forms[47].

[47] There is **NO "manual upload table" for vendors**. That feature is only for beneficiaries.

There are three key highlights based on comments from testers: (1) inform all agents that the geographical locations asked about the vendor are for where he/she is a merchant – not where the vendor was born. (2) The GPS coordinates are optional. (3) There is a hidden field for the T4 in the database table for an admin to input in, once finance produces the number.

The Fair



Preparations for the Fair

- The first point of preparation is practicing to feel confortable with the tools.
- **Make sure you prepare to create areas of shade, buy tents if needed.**
- Make sure the settings from above are set and **the devices are charged**
- Start raising awareness[48] in the community about all fair details.
- Make Beneficiary Cards and distribute them ahead of time. During awareness raising campaigns, highlight the importance of the card.
- Make Vouchers
- Print Cards and Vouchers
- Have T-Shirts for staff, signs for visibility with logos, etc.
- **Bring a couple paper receipt booklets**
- **Bring whole punchers for all voucher distributors**

[48] With CRS and Caritas staff, hired as animators in the communities, or radio ads, or sights, etc.

Making Beneficiary Cards

This process was briefly discussed above with the employee card. However for the beneficiary identification cards, **make sure the QR codes are linked to the "household ID numbers,"** and not a long text string with lots of data. The simpler the barcode, the faster the barcode reading time is, and this can make a difference when you have a line of 500 people. Making cards is easy, but please look into an already existing Bartender guide.

1) Make your ID template, or download the latest DRIVE DRC Beneficiary ID template.[49]
2) Download or make an excel file with beneficiary information, for example the table from the beneficiary registration form database, or the version in the mini server.
   - Make a column with the "photo filename" in the DRC Dropbox folder.[50]
     - The equation cut the hyperlink to the picture in Dropbox is: `=MID(G2,FIND("field", G2),LEN(G2))`

       Replace the **G2** with the right cell name.

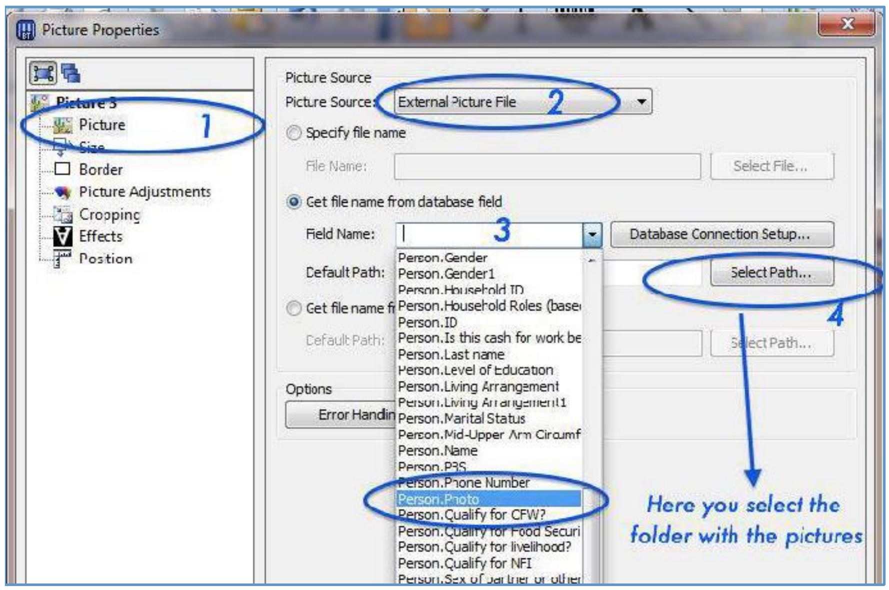
     - Drag the equation down to the last record to extend the equation all the way down to the last record.
   - Save the excel file, and link this file as a bartender database (see #1 here).

[49] The latest version reflects changes mentioned during the last DRIVE fair forms review.

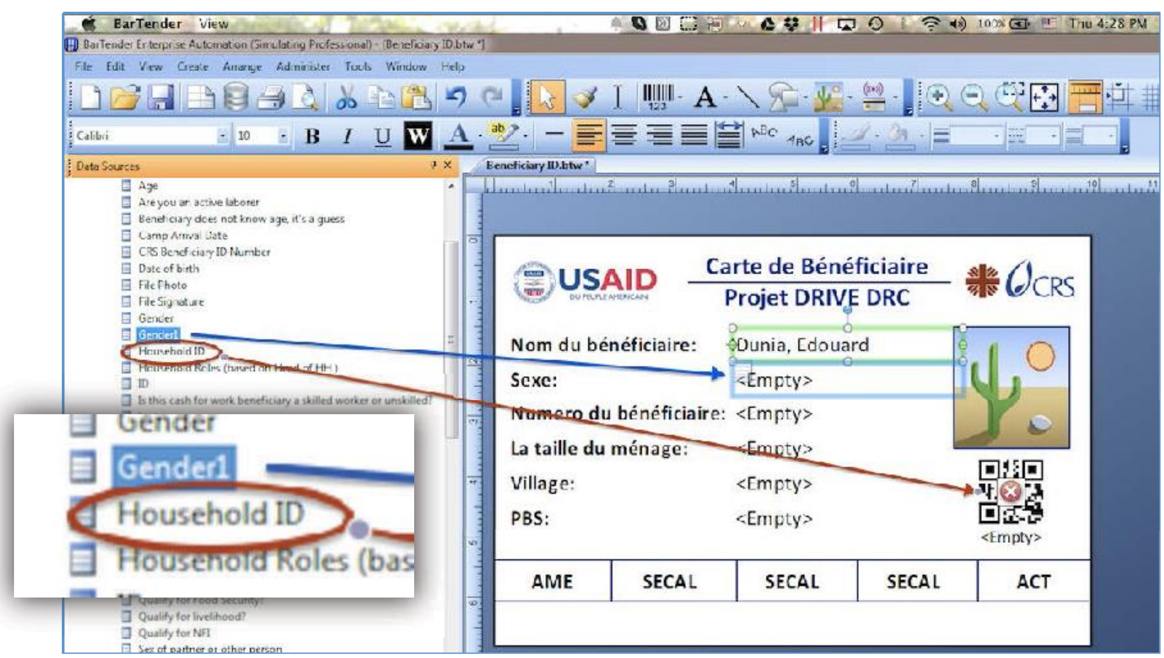
[50] IFormbuilder is set up to download all collected pictures and signatures into a Dropbox folder (crsdrcmeal.ict4d@gmail.com). The MEAL DRC Dropbox account was installed in the database officer's PC.

3) In bartender, open the ID template and set the "picture holder" (cacti) element to the new column from the excel file.

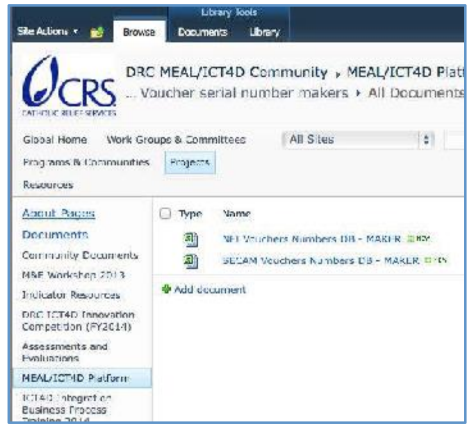
- Double click on the photo placeholder and change the element to the following settings. (#3 is "photo filename")



- Bartender is all about dragging and dropping

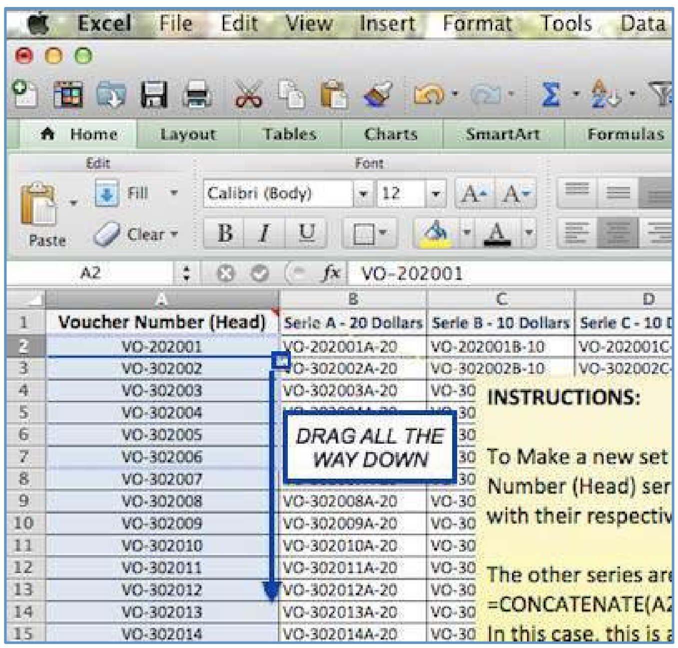


Making Vouchers



Making vouchers is similar to ID cards.

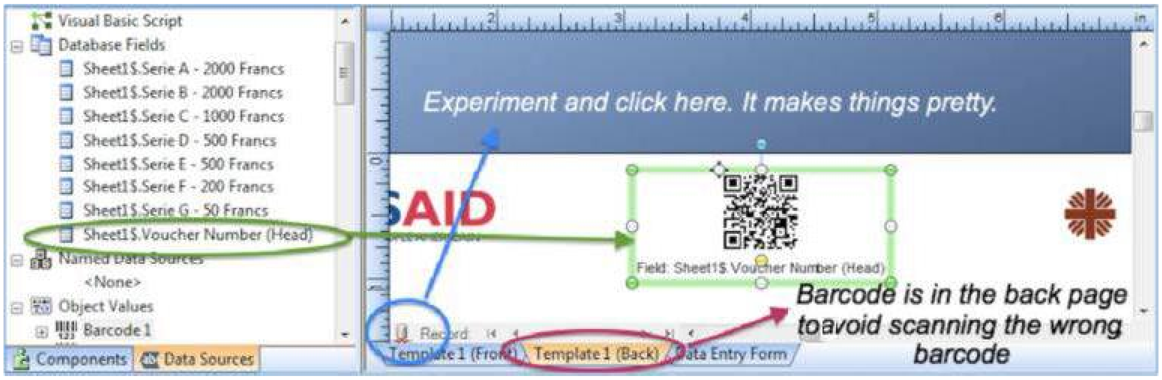
1) Download the right voucher database maker files; there are two, one for NFI and one for Food Security.
    a. They are easy to use; you choose the first number in the series in the first column, and drag all the way down. The series in the spreadsheets has not yet been used[51].

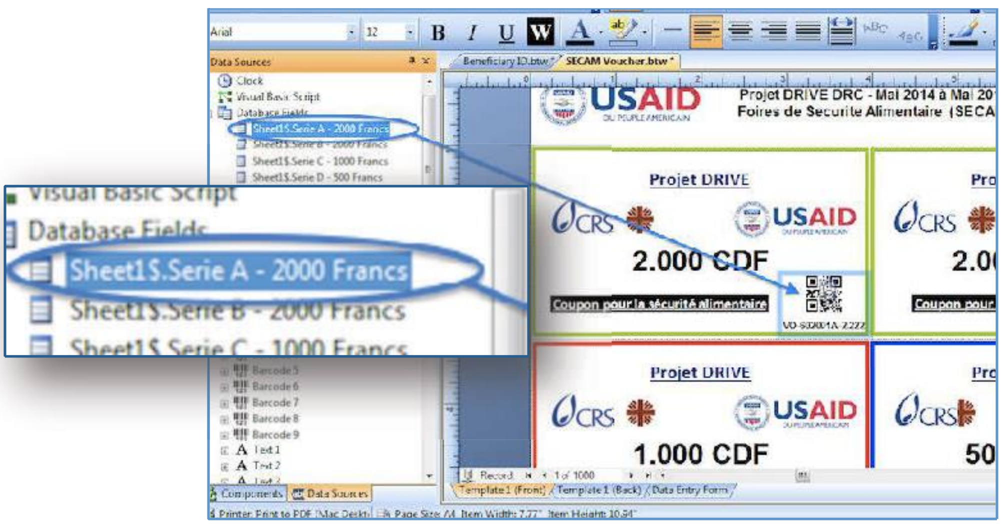


2) Download and open (or make) the voucher template. There is one for NFI and one for Food Security already made.
3) Link your new voucher database (excel file) to bartender.

[51] As of July 29, 2014

4) Link every barcode to its respective barcode. Make sure you assign the main barcode to the Voucher head column. Then series A for the first sub-voucher, then B for the second… and so on.





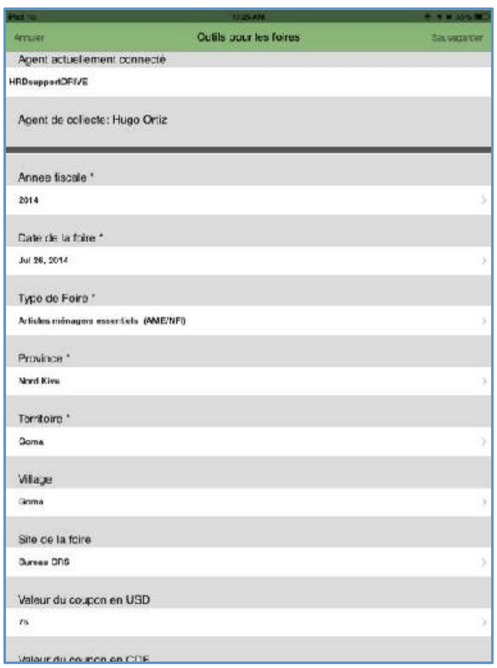
Barcodes may need to be rearranged after linked to a column.

5) Once you have linked every barcode to its column, look at a couple pages to make sure they all work, then you can print them. Bartender prefers to print from the same computer to ensure the barcode will work, but you may export the file onto a PDF, since simple QR codes should work with high resolution printer, just make sure you test, before you print a 1000 pages. **Finally, the vouchers should be in thick but perforated paper, to make the vouchers easy to tear and separate.**

The fair form

The fair form is actually many forms built into one suite of tools. The concept is to fill in the parent form with the static information **one time** (information that won't change for today). Consequently the sub-forms become "the tools" assigned to each agent that will never ask for that static information again.[52]

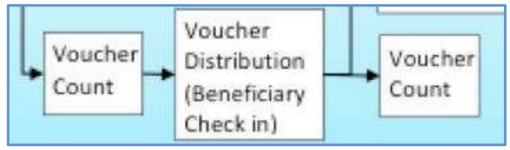
After you fill out the basic information, you will see the suite of fair forms in the following order: (1) **voucher tracking** forms, (2) **beneficiary check-in** form (and voucher number assignment log), (3) **vendor tools**,[53] (4) **beneficiary exit survey**, (5) **vendor exit survey**, and the (6) **complaints** forms. Please note that the parent form sets a lot of the skip logic for other forms. Thus, it must be filled out before using the sub-forms.



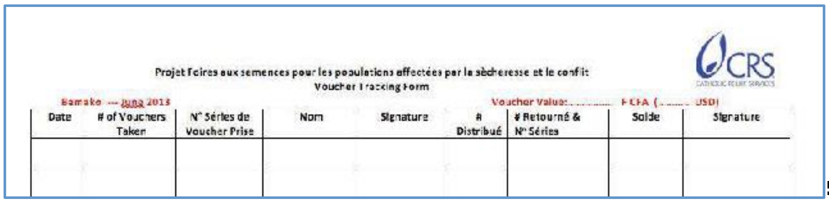
[52] The exception being if you want to send finance forms to the mini-server, since you have to send in the parent to do so. But the parent can also be duplicated once filled out.

[53] This suite includes the **vendor checking form**, **inventory** logger and **tracker before the fair**, **inventory tracker after the fairs**, automatic **voucher calculation** form, and **vendor payment** forms.

Voucher Tracking Form, before



This form allows for accountability with vouchers distribution agents. Just like a paper equivalent, you write down how many vouchers were supplied to the distribution agent before distribution, and compare that to the count (if any) after.



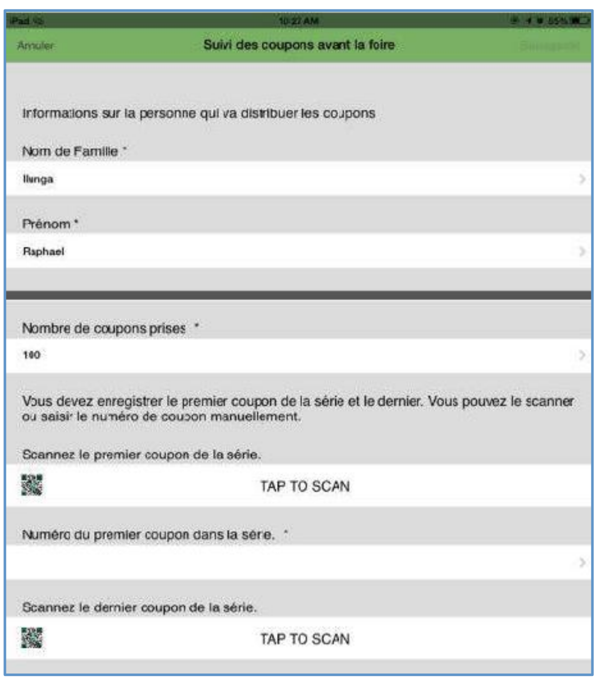
Projet Foires aux semences pour les populations affectées par la sècheresse et le conflit
Voucher Tracking Form

Bamako — Juna 2013 Voucher Value: .... FCFA (........ USD)

Date	# of Vouchers Taken	N° Séries de Voucher Prise	Nom	Signature	# Distribué	# Retourné & N° Séries	Solde	Signature

[54]

With the digital version, this one table was spilt into two sub-forms, **before** and **after.** But these forms just like its paper cousin, **requires vouchers to be printed and supplied in sequential order.**



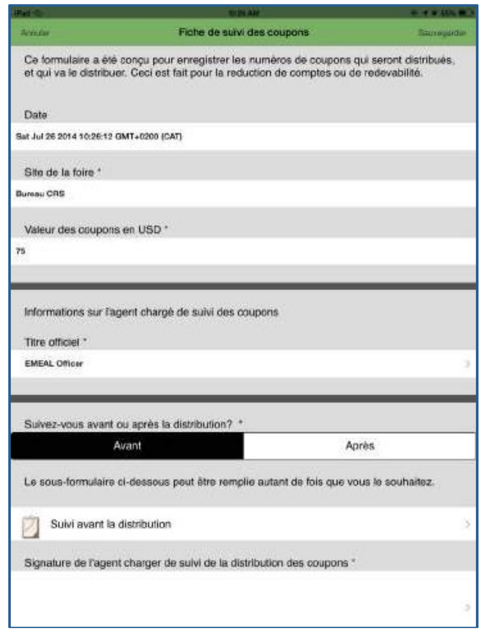
And having two makes digital collection easier.[55]

---

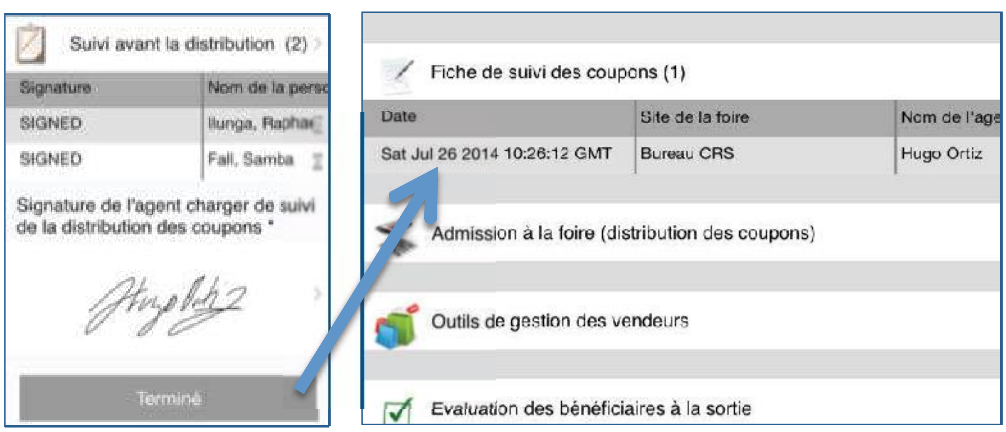
[54] A picture of a paper "voucher tracking form" from Mali

The following are the steps for the "CRS agent" tracking vouchers before the distribution:

1) Register the static information at the "parent" levels.[56]
2) Open the "**before**" sub-form and have the distribution agent type in their name.[57]
3) Write down how many voucher the first person receives and which one [58] (by scanning or typing).
4) Capture agent's signature and move on the next agent (open the "before" sub-form again and repeat process with the next agent).



5) Once the information was collected for all voucher distributors, we finish the main voucher tracking form and can move on to help at the next activity, while the agents distribute vouchers.



[55] And more compliant with the needs for Finance than the original paper forms, e.g. title of tracker.

[56] The "voucher tracking" is a child of the parent "fair tools form." But "before" is a child of "voucher tracking," i.e. the other parent.

[57] While not a big problem if it doesn't happen, if the name matches in both the before and after forms, the form does automatic calculations.

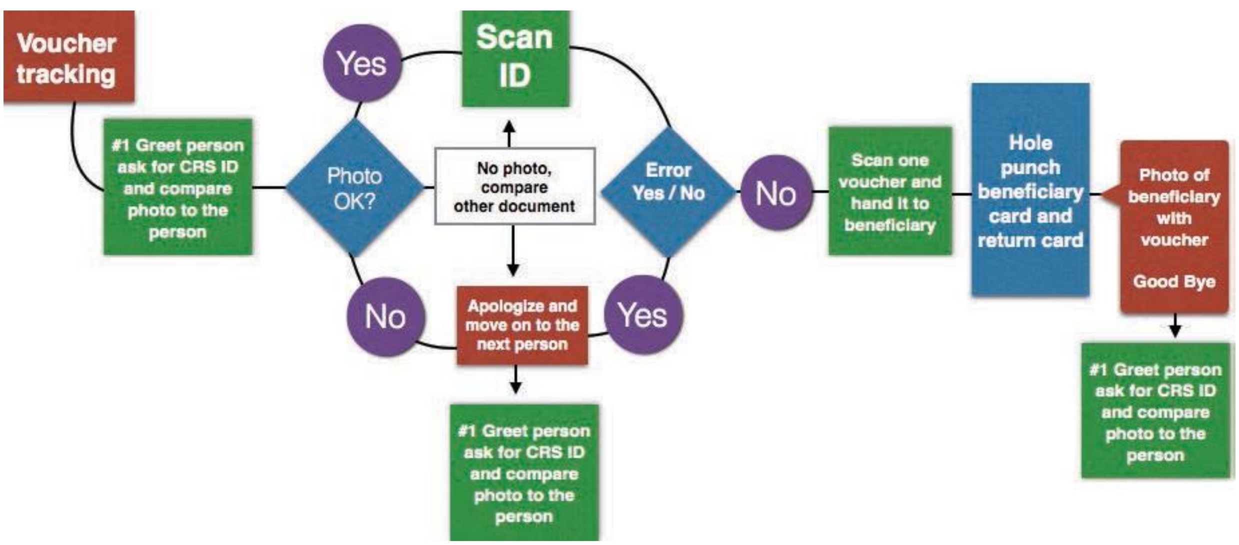
[58] **The voucher (HEAD) number should be scanned, not the small one in each sub-vouchers.**

Beneficiary check-in

The team managing the beneficiary check-in form does a lot – i.e. (1) they record the time and date the beneficiary receives their voucher, (2) they record all the information about the beneficiary (name, sex, village, household size), (3) they record the voucher number and forever link it to the household ID number, they (4) capture proof[59] that the beneficiary received the voucher, they (5) tag the beneficiaries as they receive benefits, and they (7) make sure only beneficiaries who qualify for vouchers, get them – all in less than a minute.

Distribution agents scanning cards

The beneficiary check-in process

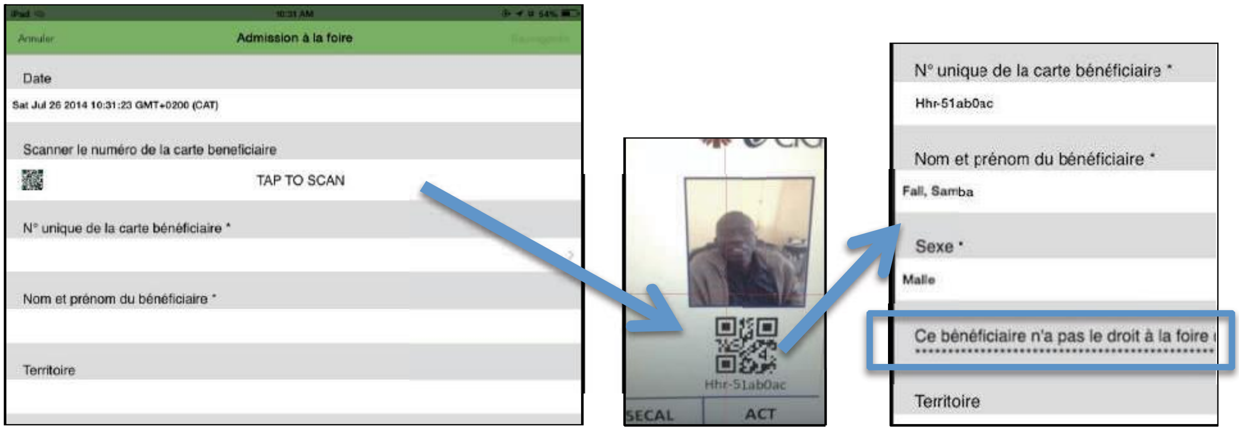


1) After greeting and comparing the photo, the voucher distributor just taps on the first element to scan the ID card. If he/she gets an error message, they quit the sub-form and move on to the next person.[60]

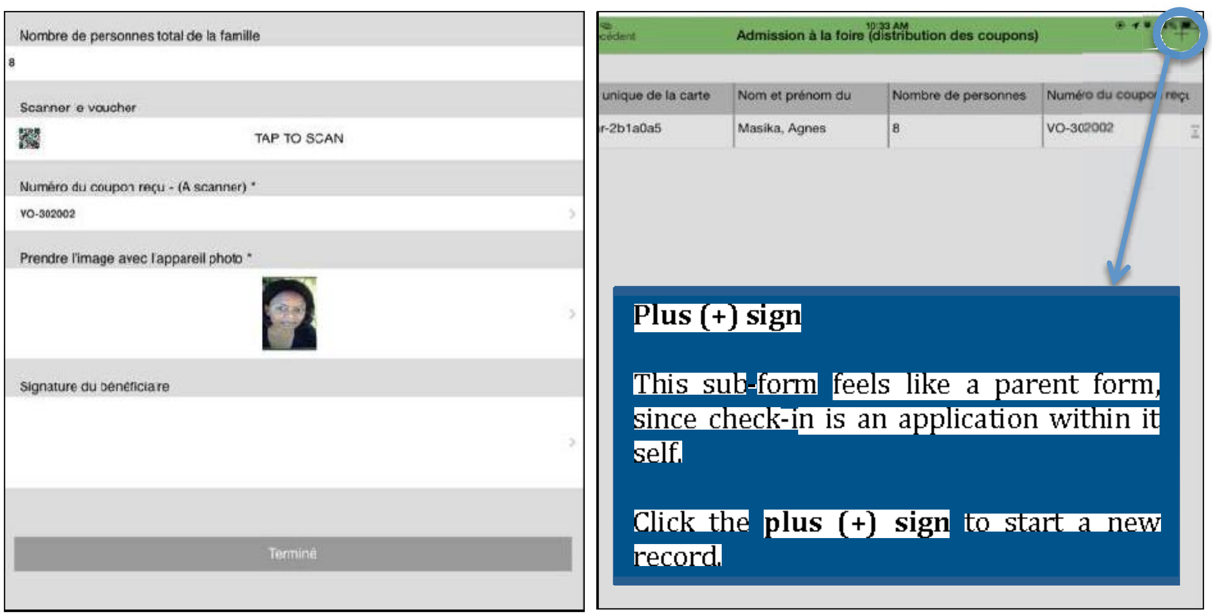
---

[59] Without controversial biometric fingerprints, instead it's just a picture or a signature or both.

[60] They can blame the tech, even if they know the person in front is not being honest.



2) If the agent scans and doesn't receive an error message, they can scan a voucher and give it to the beneficiary. Then the beneficiary card is hole punched (in the right box) and returned to the beneficiary. To get proof of distribution, we then take a picture of the **beneficiary holding the voucher**, or alternatively the beneficiary may sign.

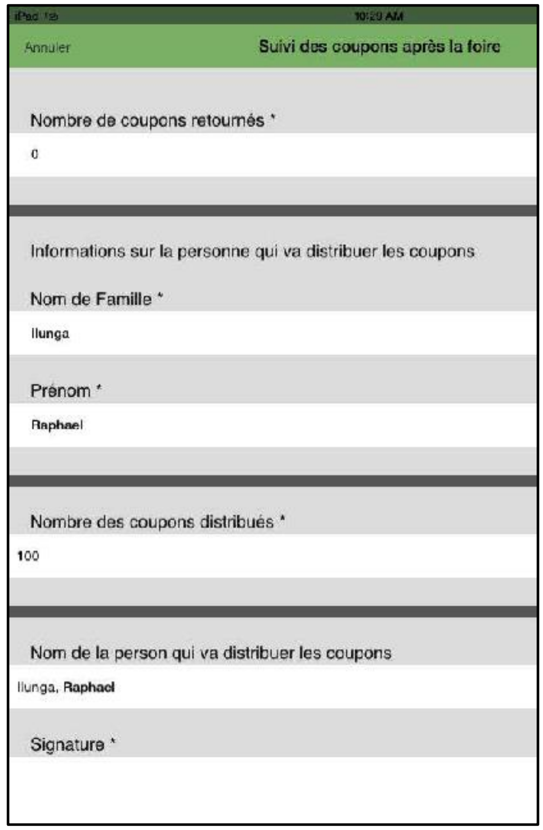


Once a distributor finished checking in one person, they start a new record (+) to start the process[61] again.

[61] **The check-in** data table houses pertinent information to fairs analyses in one file, i.e.: (1) household (HH) size, (2) breakdown of HH members by sex, (3) village, (4) territory, (5) province, (6) sex, (7) vulnerability status, (8) whether beneficiary received health kit, and more. However, if something is missing, all data is linkable if there is a **unique token ID**, like the household ID or beneficiary ID.

Voucher Tracking Form, after

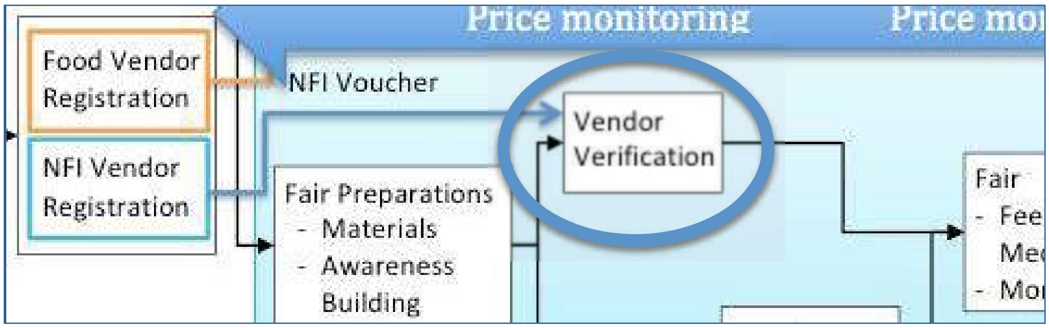
This part is straightforward. Here you should know the math is automatic if the name of distributor is written with the same spelling as the previous time, but either way the important thing is to (1) record the number of vouchers returned, (2) the total value returned, and (3) a signature. The first question asks for the number of returned vouchers. If the distributor is returning 0 vouchers, the form won't ask for the serial numbers of the vouchers received.



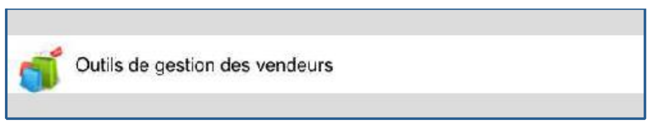
Finally, in the days of databases, relational tables, Zoho reports and other stuff like that, one would not be concerned about having multiple tables for something. After all, we could link data from any one table to any table with a token ID number.[62] However for the nostalgic, *all the information from the paper cousin (voucher tracking form)* ***comes out in one table*** – in the "**after fair voucher tracking**" table.

[62] As long as there is a common linking element, like a household number any form can be relational to another. This can be done in Zoho Reports, File Maker Pro, Bartender, Microsoft Access, and many more…

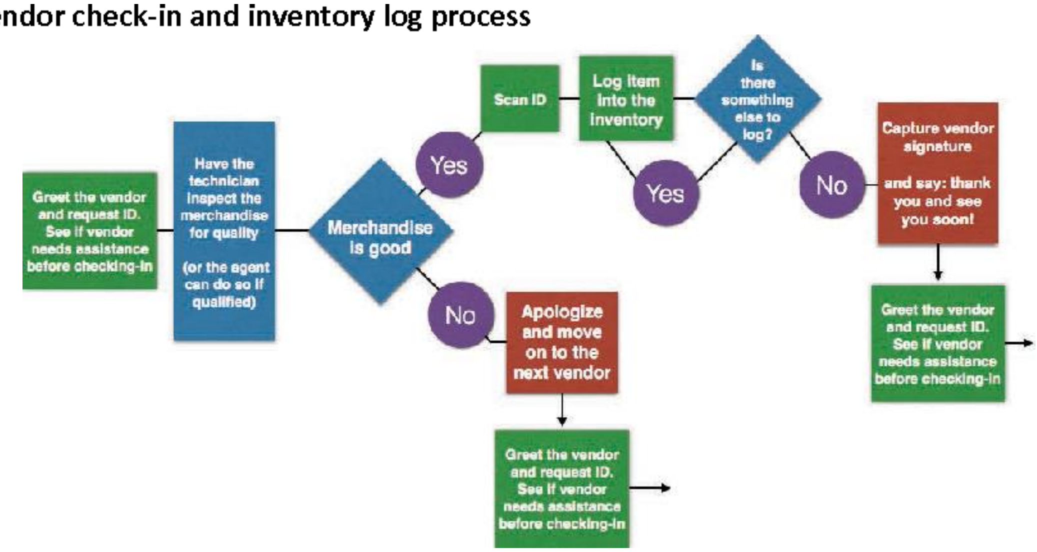
Vendor Check-in (Verification)



A team of agents will be in charge of working with vendors. Once a vendor arrives at the site of the fair, his time of arrival has to be recorded. Also, the merchandise brought has to be checked[63] and logged into an inventory.



The vendor check-in and inventory log process



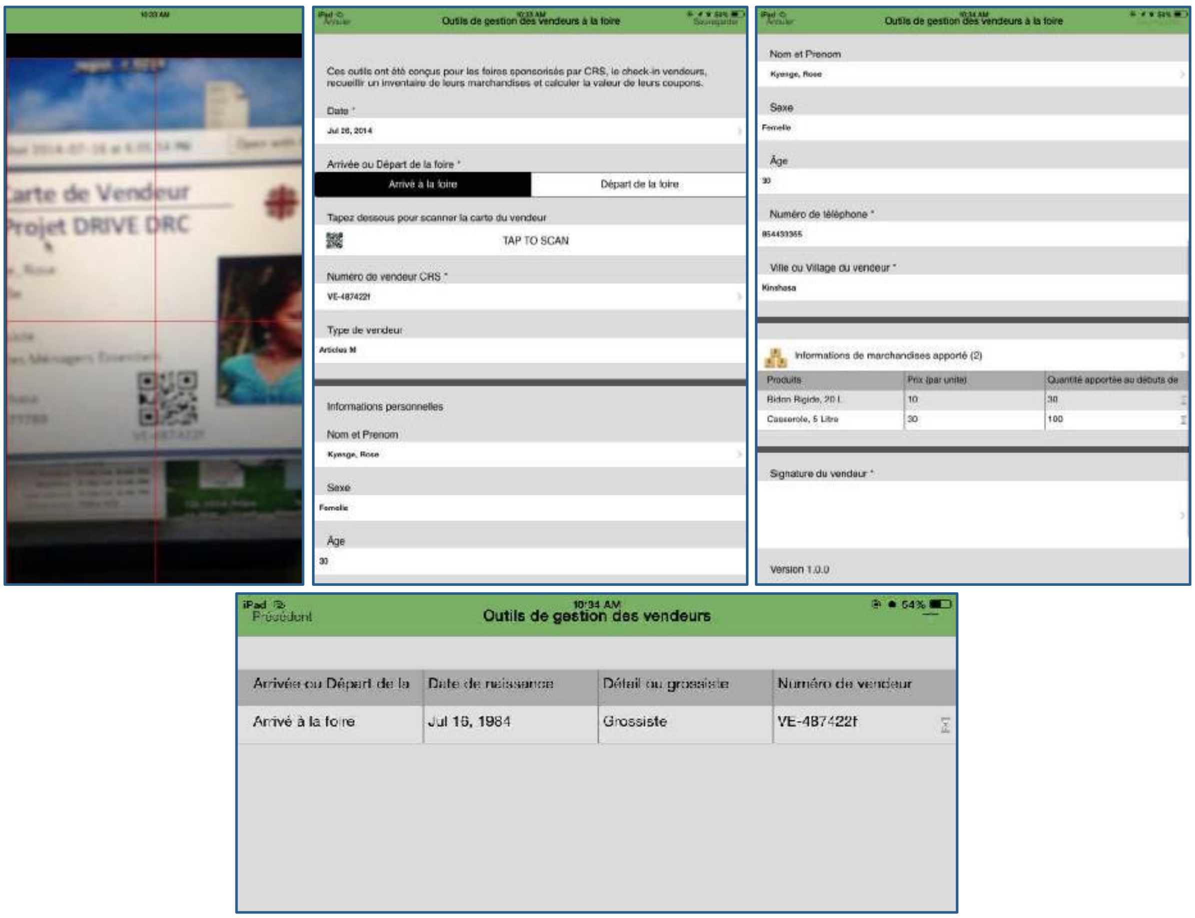
1) The first thing to do is to be polite, say hello and offer assistance if needed. Once the vendor is ready to be checked in, (1) open the vendor sub-form[64] (or start a new record), (2) take the ID card, and (3) have a trained technician check the

[63] Make sure of the quality is right, but also that the promised merchandise is the one that was brought in.
[64] The form asks if you are using the tool before or after the fair, select "**before**."

merchandise. Once it's confirmed that the merchandise is okay, we can scan the ID to check-in the vendor at the fair.

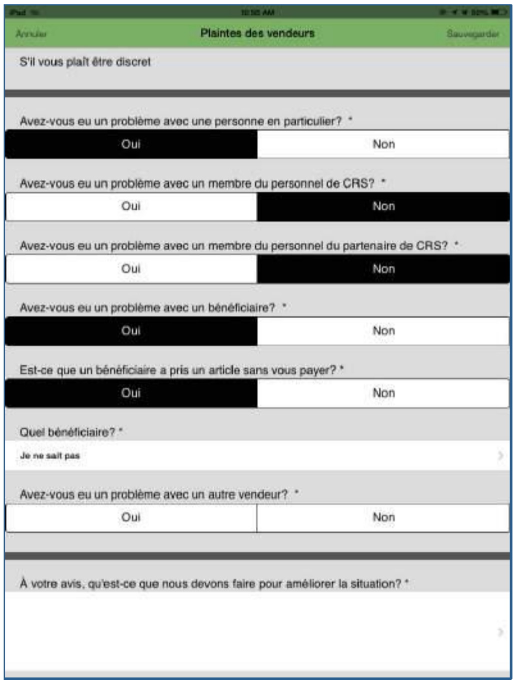
2) Then, we have to fill in the inventory sub-form to log all the merchandise items brought into an inventory database. For each item type (not quantity), the form must be filled out. Once done, you complete the vendor tools sub-form to finish.

3) **Then, inform the vendor he/she must check out <u>with the same agent.</u>**

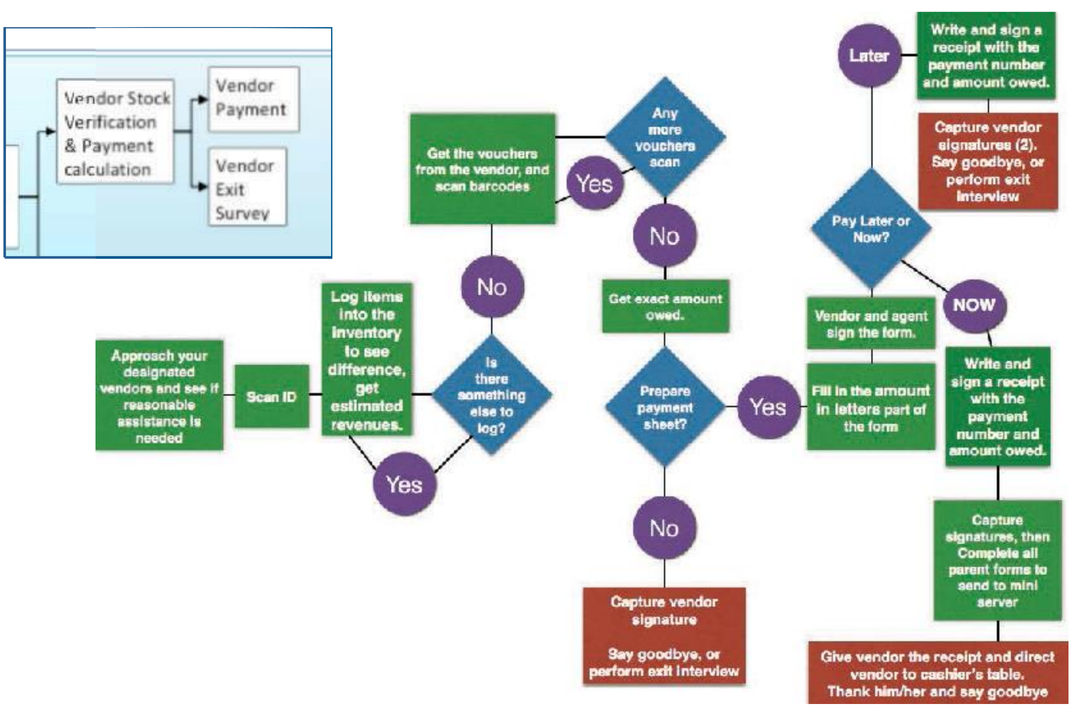


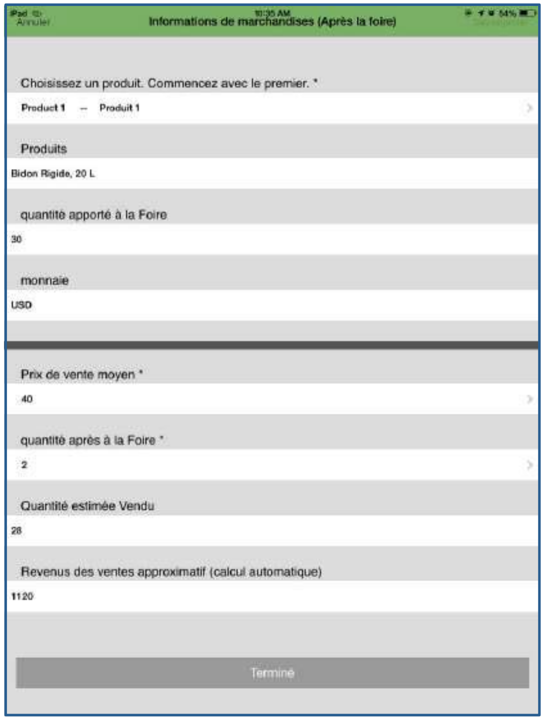
Feedback at the fair (Complaints)

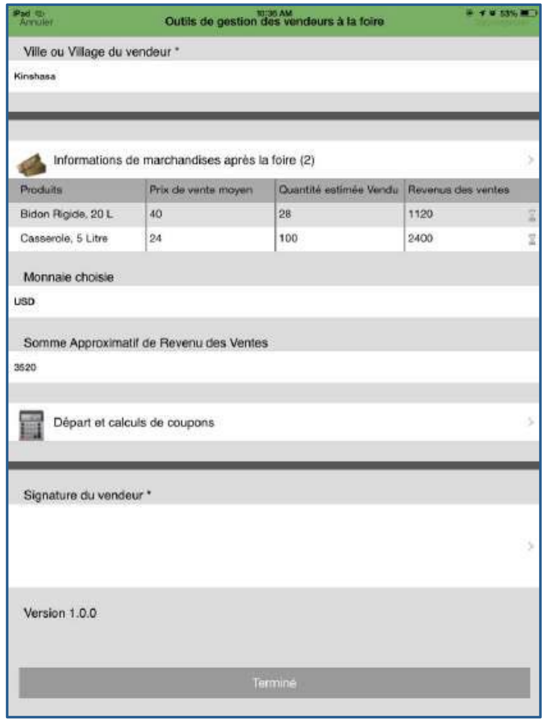
Most of the feedback from beneficiaries will be collected during the exit interview, however during the fairs (or before and after), there might be problems for which beneficiaries and vendors want to complain about. The complaints form was divided into two sub-forms, vendors and beneficiaries. Initially the form was very open and depending on some answers, it still is. *Gregoire Mukenge wrote most non-open questions.*

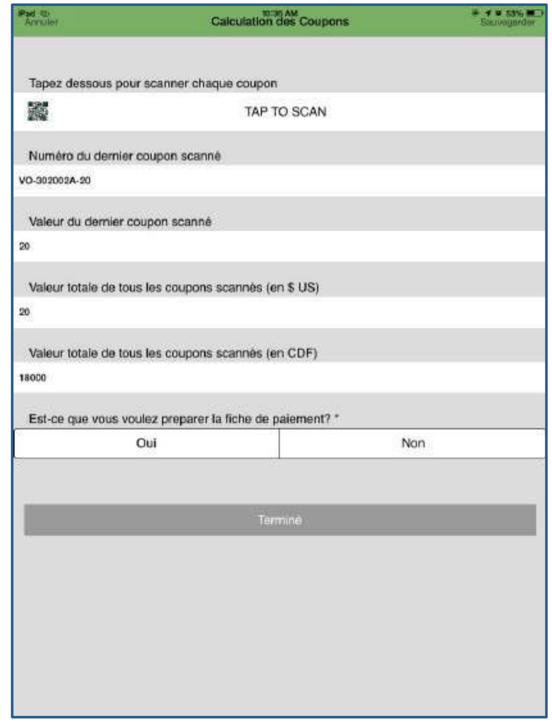


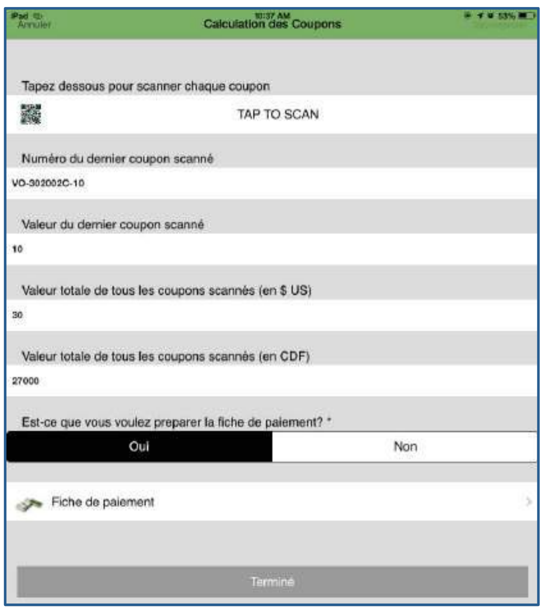
Vendor tools: Vendor stock verification and payment calculation

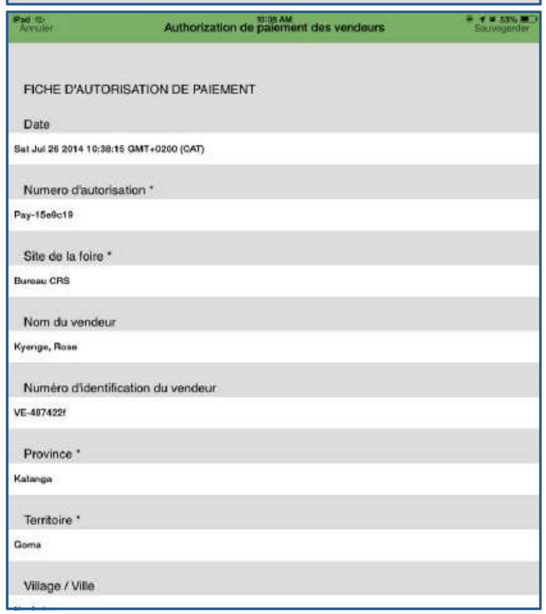


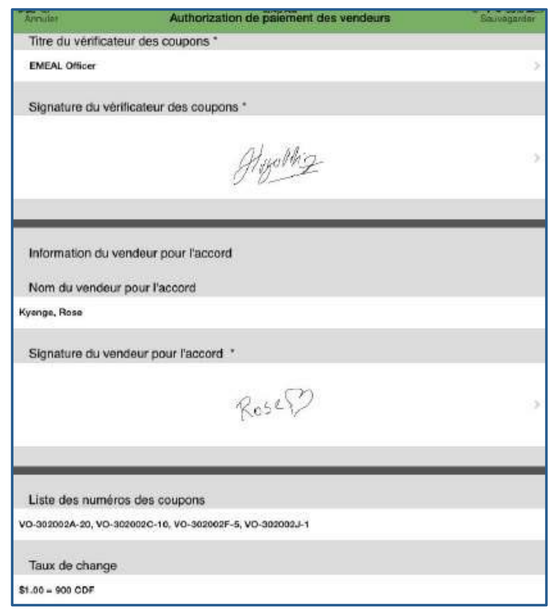

Annuler
Informations de marchandises (Après la foire)
Choisissez un produit. Commencez avec le premier. *
Product 1 – Produit 1
Produits
Bidon Rigide, 20 L
quantité apporté à la Foire
30
monnaie
USD
Prix de vente moyen *
40
quantité après à la Foire *
2
Quantité estimée Vendu
28
Revenus des ventes approximatif (calcul automatique)
1120
Terminé


Annuler
Outils de gestion des vendeurs à la foire
Ville ou Village du vendeur *
Kinshasa
Informations de marchandises après la foire (2)
Produits | Prix de vente moyen | Quantité estimée Vendu | Revenus des ventes
Bidon Rigide, 20 L | 40 | 28 | 1120
Casserole, 5 Litre | 24 | 100 | 2400
Monnaie choisie
USD
Somme Approximatif de Revenu des Ventes
3520
Départ et calculs de coupons
Signature du vendeur *
Version 1.0.0
Terminé

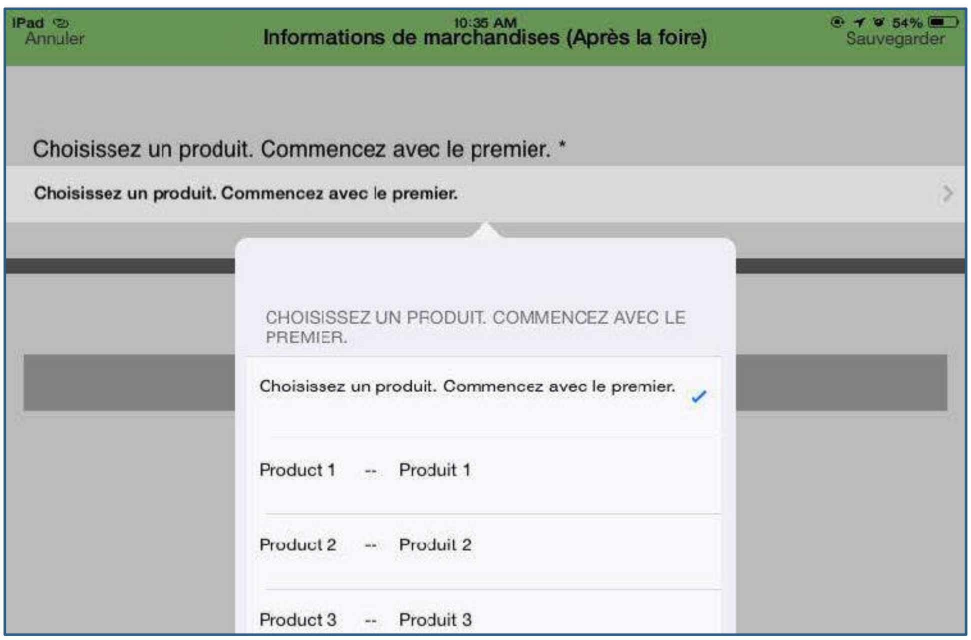

Annuler
Calculation des Coupons
Sauvegarder
Tapez dessous pour scanner chaque coupon
TAP TO SCAN
Numéro du dernier coupon scanné
VO-302002A-20
Valeur du dernier coupon scanné
20
Valeur totale de tous les coupons scannés (en $ US)
20
Valeur totale de tous les coupons scannés (en CDF)
18000
Est-ce que vous voulez preparer la fiche de paiement? *
Oui
Non
Terminé


Annuler
Calculation des Coupons
Tapez dessous pour scanner chaque coupon
TAP TO SCAN
Numéro du dernier coupon scanné
VO-302002C-10
Valeur du dernier coupon scanné
10
Valeur totale de tous les coupons scannés (en $ US)
30
Valeur totale de tous les coupons scannés (en CDF)
27000
Est-ce que vous voulez preparer la fiche de paiement? *
Oui
Non
Fiche de paiement
Terminé


Annuler
Authorization de paiement des vendeurs
Sauvegarder
FICHE D'AUTORISATION DE PAIEMENT
Date
Sat Jul 26 2014 10:38:15 GMT+0200 (CAT)
Numero d'autorisation *
Pay-15e0c19
Site de la foire *
Bureau CRS
Nom du vendeur
Kyenge, Rose
Numéro d'identification du vendeur
VE-487422f
Province *
Katanga
Territoire *
Goma
Village / Ville


Annuler
Authorization de paiement des vendeurs
Sauvegarder
Titre du vérificateur des coupons *
EMEAL Officer
Signature du vérificateur des coupons *
Information du vendeur pour l'accord
Nom du vendeur pour l'accord
Kyenge, Rose
Signature du vendeur pour l'accord *
Rose
Liste des numéros des coupons
VO-302002A-20, VO-302002C-10, VO-302002F-5, VO-302002J-1
Taux de change
$1.00 = 900 CDF

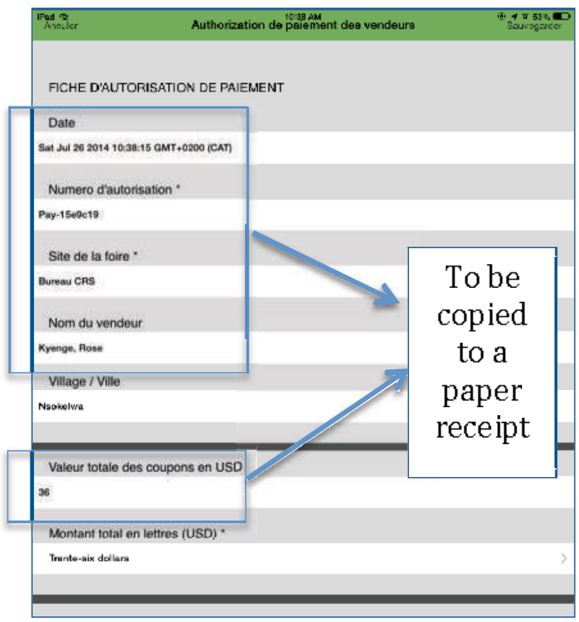
1) To checkout vendors, we fill the same form i.e. vendor tools but answer “after” to the before or after question.
2) You scan the vendor card, and proceed to the “merchandise after fair” sub-form.
3) In this sub-form, the first element asks you to choose a product number, the first is the first product that was logging into the inventory before the fair, the second is the second, and so on. *(Again, for this to work in an offline environment, the agent that checked-in the vendor must check him/her out).* Register the change of all products that were logged earlier. This gives estimated revenues for each product sold based on average prices at the fair.



4) Once you’re done logging inventory, you proceed to the voucher calculation. Scan each voucher and just watch how they automatically get counted and their values summed. Make sure the sub-form worked on the first scan, if not – quit the sub-form and start again.

5) If you decided to generate a payment form, it will be automatic except for "the numbers in letters" text field. You may copy all the information on the screen to a paper form, or you may finish all forms including the parent fair form to send it to the mini-server.[65] To sync your device with the mini-server, drag the screen down[66] (A) or go to the settings menu (B).

6) In both scenarios, (1) if you pay the vendor at the fair or (2) if you pay the vendor in the future, a **simple paper receipt should be filled out and handed to the vendor.** The receipts should have the payment authorization number, the date, the name of the vendor, and the amount owed. If the vendor is to be paid that same day, he/she is then to proceed to the finance or cashier table.



[65] One drawback of sending the form to the mini-server is that the parent fair form must be completed to send. However, one strategy to save some life seconds (*only for agents checking out vendors*) could be to fill out the fair parent form one time. Save it and duplicate it once. Once the first form is gone, you use the duplicate (and make a duplicate of the copy before using it). **To duplicate a record, just tap and hold on the record.**

[66] One must have completed the parent sub-form and be outside of it to drag screen down.

Finance table (if paying at the fair)

1) Once the vendor payment form has been filled out and all the parent forms were finished and synced, the vendor should take his/her receipt to cashier table to receive payment. This was done this way because it is a good practice to have a separation of tasks for accountability and to avoid fraud.
2) The cashier needs a laptop computer and a portable printer. He/her must join the mini-server's Wi-Fi network. A Voltaic solar panel or other power source to provide the electricity for all these things might be needed as well.

3) To log on to the mini-server to download the payment forms, the cashier and project manager would use their web browser (Chrome, Safari, Firefox, or Internet Explorer) to go to the mini-server's URL: **https://192.168.40.100/exzact/dataViews.php** (this address only works within the mini-server Wi-Fi network, it is an **Intranet**).
4) The cashier must then log on with the **DRIVEcashier**[67] account; this account only has access to view one table, making things easy, i.e. payment authorization form.

[67] There is a repository with all passwords for the DRC. This and all other passwords should be stored over there.

5) Click on the list, then click on the **"List"** button, or simply click on the ID number for the table (look at screenshot above). You will then see a small database with all the completed payment forms. You can search for the one you're looking for by comparing authorization number and/or name of vendor. Click on the form that you want to download and print, and then click on the "**View Detail"** button.

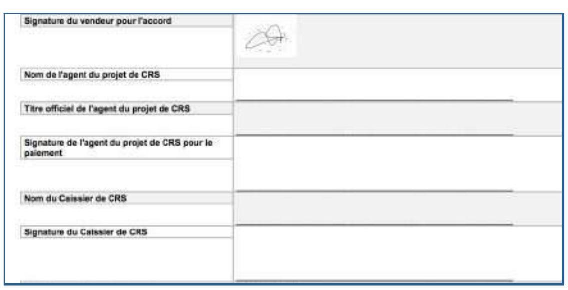
Authorization de paiement des vendeurs

Select All | View Detail

Date	Numéro d'autorisation	Nom du vendeur	Site de la foire	Numéro d'identification du vendeur	Province	Territoire
Fri Jul 25 2014 14:18:28 GMT	Pay-da29388	Kyenge, Rose	Ouest	VE-487422f	Katanga	Pweto
Fri Jul 25 2014 14:27:18 GMT	Pay-293f312	Bemba, Pierre	Ouest	VE-cebd2a3	Katanga	Dilolo

6) You will then see all the information from the payment form, but click on the PDF button to make the file that will be printed.

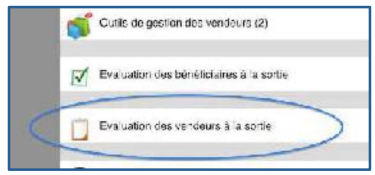
7) This file can then be opened with "Adobe Reader" or any PDF viewer. **Open it and print it.**

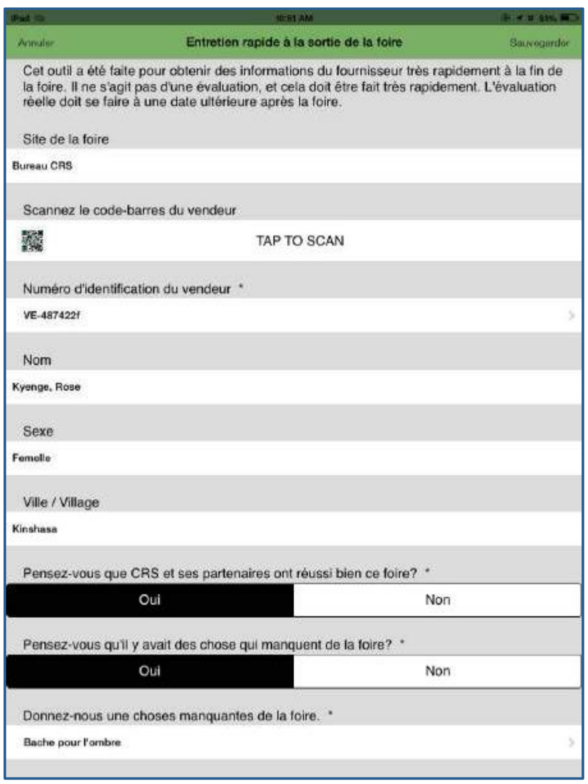


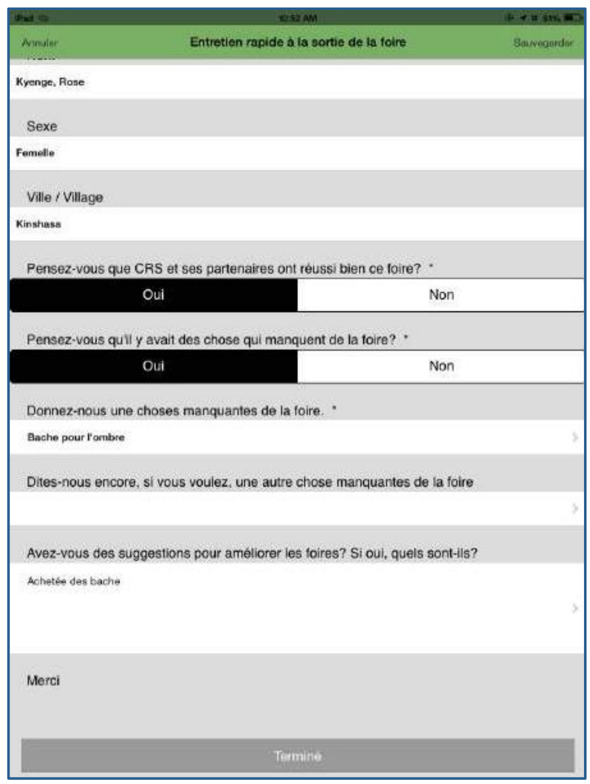
8) Then have both the cashier and third designated person (usually the PM or PM delegate) **sign and date the form with pen** (look above, the form leaves blank spots for signatures). This enables the existence of "an original" and a paper trail.

Vendor exit survey

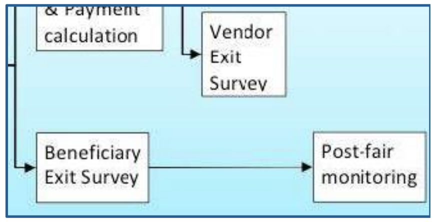
During the mapping exercise (June 4th), it was concluded that the best feedback from vendors was to be collected a few weeks after the fair. This is because at that point, one can synthesize a complete understanding of the vendor's experience across multiple fair dates and operational tasks (including payment). However a quick exit survey was built in to the fair tools, to get feedback from a representative sample of vendors immediately after the fair. This should be able to be completed in less than 1-2 minutes; it does have some open questions – **but they are not required.** This sub-form is located within the "fair tools" parent form, not the "vendor tools" sub-form.

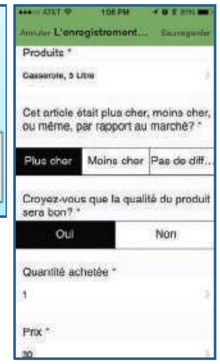




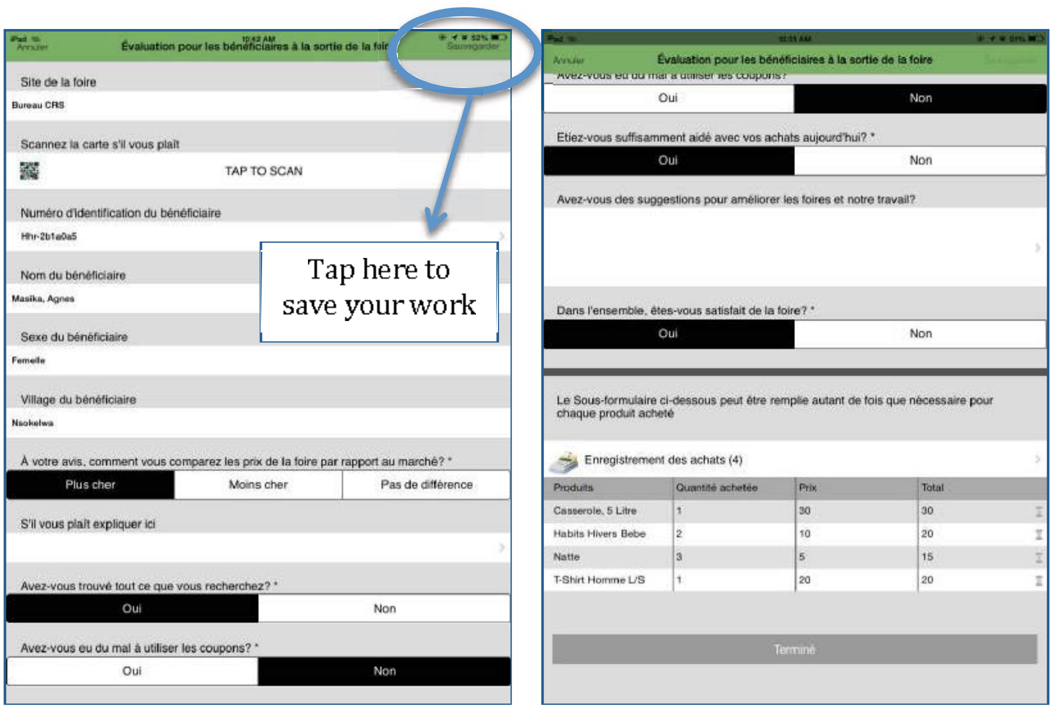


Beneficiary exit survey





Just like the vendor counterpart, this form is direct and meant for a representative sample as opposed to the whole population of beneficiaries that attended the fair. However, **this form also has a sub-form to record each purchase the beneficiary made.** It records the (1) item's type, (2) price per quantity, (3) quantity, and (4) perception of quality. This form also has non-required open questions.



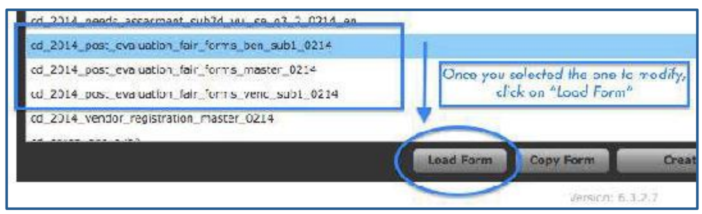
The last step, one tip, plus what's missing.

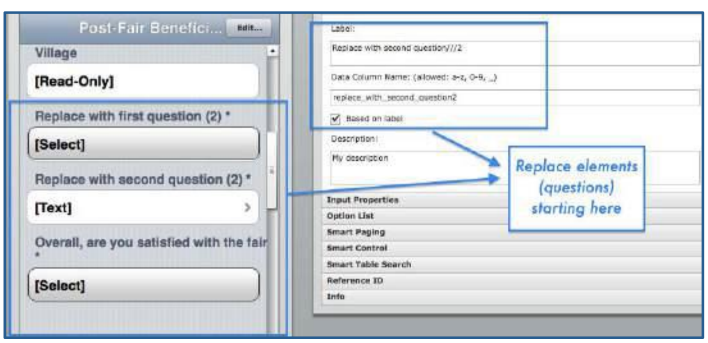
The last step for all apps in the "fair tools" is to finish the parent form and sync the iPad. For most fair tasks, this will only be done once (the exception being if you pay vendors on the spot, where you can copy the parent).

The tip is to have people "save their work" constantly. While iForm saves automatically every time you finish a sub-form record, you never want to loose what you did for the last 2 minutes if the app crashes.

There are missing forms (questionnaires that have not yet been developed) as of July 31st 2014 for "a complete fair solution." They are the **"post-fair monitoring"** evaluation forms. One is needed for both beneficiaries and vendors with skip logic to keep things pertinent depending on the type of fair i.e. non-food-items or food security.

Conversely, these forms should be easy to develop and **templates** (pre-made forms for this purpose) were already made[68] to speed their development.[69] **NOTE: because these forms are not (100%) done, they are not yet part of any DRIVE form group.** Once the forms are done and tested,[70] the parent can be set to form part of a DRIVE form group. The screenshot below is the form template within the **form-builder,** made for the post evaluation forms.



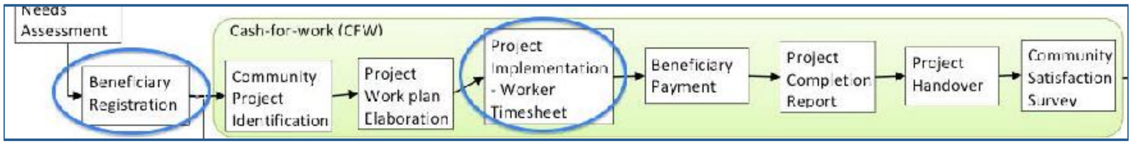


[68] There are three forms that were not listed in the report, one parent and two children sub-forms. The parent's name is: *cd_2014_post_evaluation_fair_forms_master_0214* – and the children are: *cd_2014_post_evaluation_fair_forms_ben_sub1_0214* and *cd_2014_post_evaluation_fair_forms_ven_sub1_0214*. ******Admin rights are needed to see and modify these forms within the form-builder.***

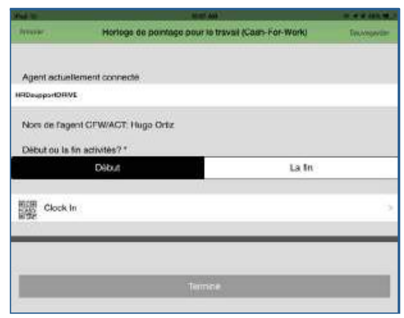
[69] What's pre done is the scanning of barcode widget and the automatic look up and display of vendor/ beneficiary information from manual upload table or (beneficiary/vendor registration tables).

[70] Look at iFormbuilder guides to learn how to change/make form groups.

Cash-for-work

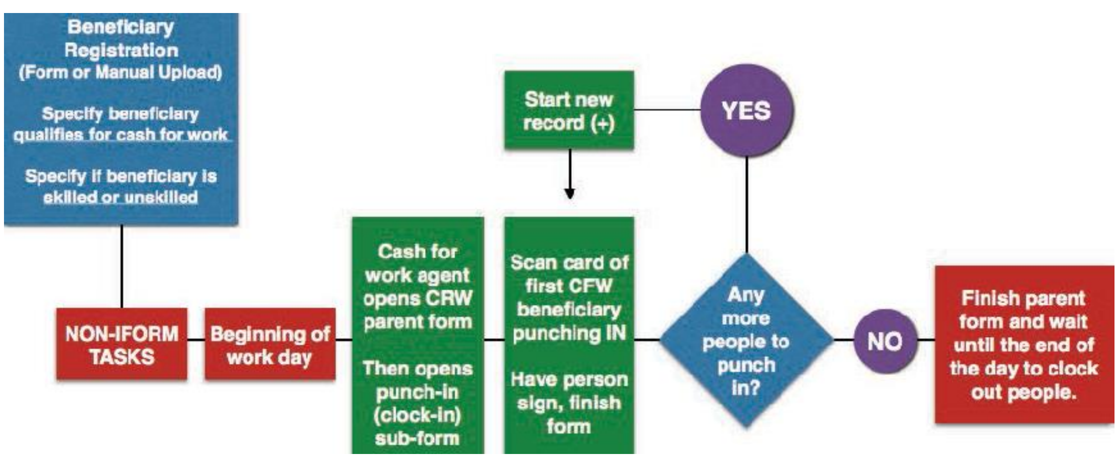


Not all tasks for cash-for-work activities require digital forms from CRS, since the beneficiaries should be the main actors helping identify worthwhile projects and designing their action plans. As of June 31st, 2014, **only one**[71] **cash-for-work** tool has been developed**, "the punch clock and time sheet."**



The punch clock logs the timestamp of when the person **starts** working that day and the timestamp for when they clock out at the **end**. At the same time, the app records (1) how many days the person has worked, and (2) how much money he/she is owed.[72]

The punch-in (clock-in) process flow

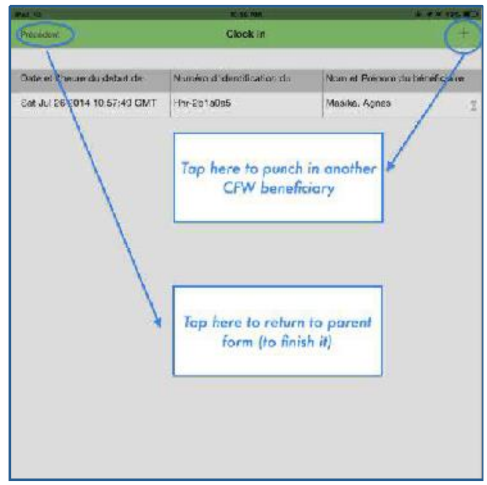


---

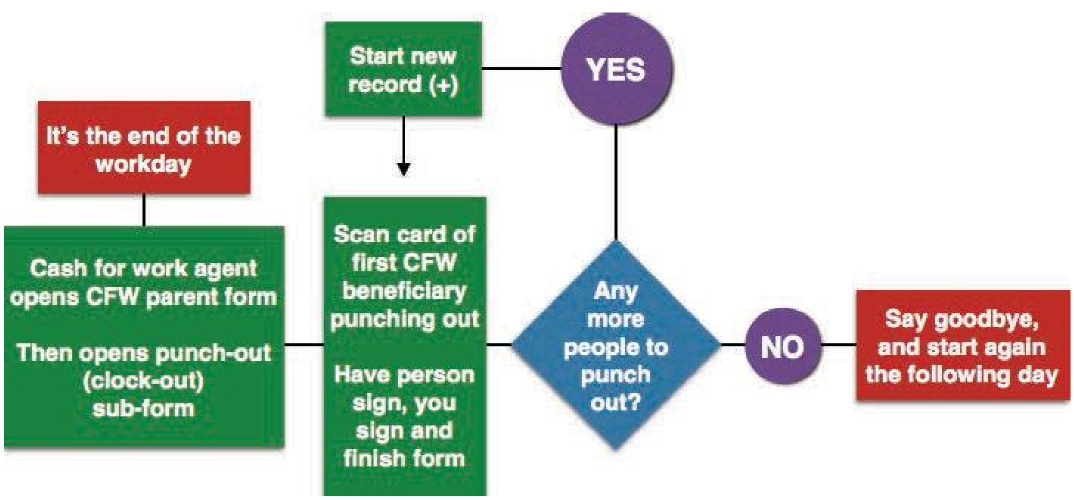
[71] DRIVE DRC team should consider making the **Community Satisfaction Survey** as a digital form. To duplicate the scanning ID card scanning functionality, the "beneficiary post-fair template form" could be duplicated. GKIM / CRS eValuate is in the process of making "public work" forms that could serve the tasks within the process map for cash-for-work.

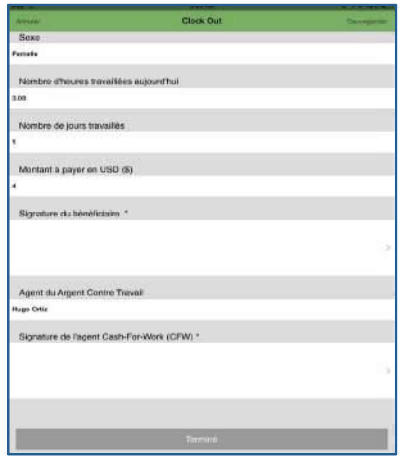
[72] DRIVE DRC makes the distinction between skilled ($4 per day) and un-skilled workers ($2 per day)

This form is very easy to use and the sub-forms act as parent forms, so as you finish with one beneficiary, you start a new record and scan the next person and have them sign. Once you are done **clocking people in**, you finish the parent form.



The punch out (clock-out) process flow

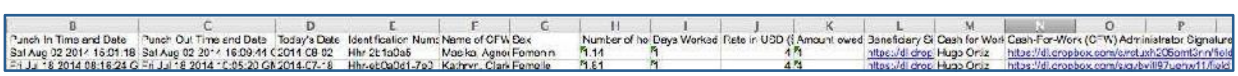




This form works the same way as the punch-in counterpart. The only thing to note is that an error message will show up if a person tries to punch out within **less than an hour** of punching in. Therefore the person can clock out, but iForm will not count this day as a workday. This was done to avoid fraud and the time frame can be changed (extended to more hours), within the form-builder.

Downloading the timesheet

While you may download the parent form's excel file (which includes everything) form "CRS iFormbuilder," the "Punch Out" table includes all the necessary information,[73] i.e. (1) punch in dates and times, (2) punch out dates and times, (3) name of CFW beneficiary, (4) his work rate, (5) amount owed up to that day, (6) signatures for CFW beneficiary and CFW officer, and (7) the name of CFW officer — (Look at screenshot of output below).



B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Punch In Time and Date	Punch Out Time and Date	Today's Date	Identification Num	Name of CFW	Sex	Number of ho	Days Worked	Rate in USD ($	Amount owed	Beneficiary Si	Cash for Work	Cash-For-Work (CFW) Administrator Signature		
Sat Aug 02 2014 16:01:18	Sat Aug 02 2014 16:09:44 (	2014-08-02	Hhr-2b1a0a5	Macka, Agnes	Feminin	1.14	1		4	https://dl.drop	Hugo Ortiz	https://dl.dropbox.com/s/ret.uxh206omt3nn/field		
Fri Jul 18 2014 08:16:24 G	Fri Jul 18 2014 10:05:20 GM	2014-07-18	Hhr-eb0a0d1-7a3	Kathryn, Clark	Femelle	1.81	1		4	https://dl.drop	Hugo Ortiz	https://dl.dropbox.com/s/q/bvill97uehw11/field		

Changing the default values (Rates for skilled and unskilled / minimum hours)

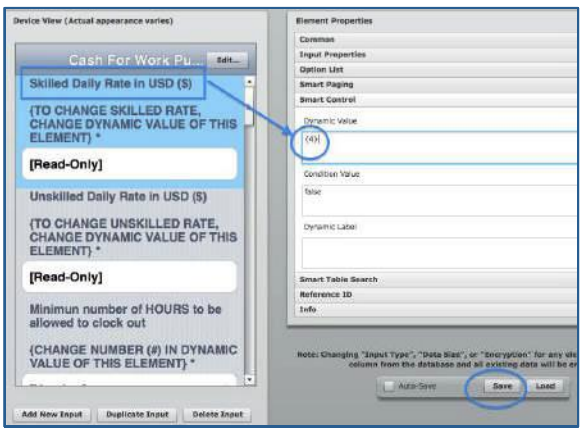
This tool (like the fair forms) was designed to be re-used by other projects, after project DRIVE (if wanted). Thus **an easy way to change the default values** for the rates and the

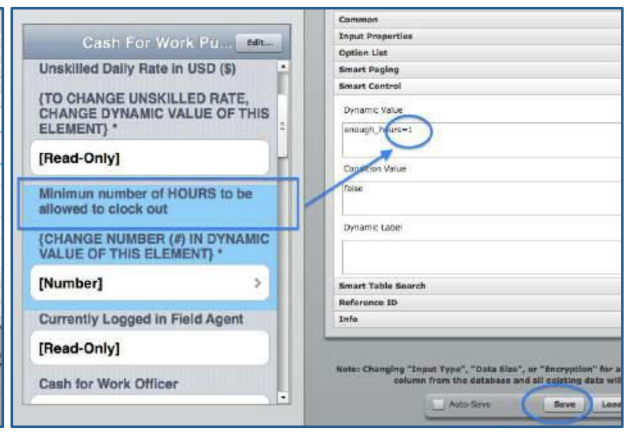
[73] Information that you would be able to find on CRS CFW paper tables and forms – its table name is: *cd_2014_cfw_timesheet_signout_sub1_0214*

minimum number of hours, was built in to the forms. However, to avoid fraud, these values have to be changed by an administrator and not within the forms.

1) Log in to https://crs.iformbuilder.com with an admin account within the profile
2) Open the form-builder, and load the master parent form for cash-for-work, the table name is: ***cd_2014_cfw_timesheet_master_0214***

3) The first three elements store these values; change by replacing their dynamic values with the value that wanted and save the form (Look at screenshots below).



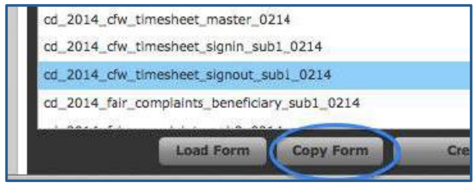


Making modifications to a form

In theory, you want to perfect the forms before you go to the field. Making modifications to a form could break other pieces of the form or even erase existing data within the table. So once a form went life, it is best to not modify it every again!

But at the same time the key to emergency response and the reason CRS is chosen to perform expensive emergency projects is because of speed and flexibility, and for this reason (and others) an existing live or non-live form might need to be modified. Therefore it's a good to have a non-destructive strategy to make modifications to forms.

1) Copy the form you want to modify to make a back-up.[74]



2) Understand that some functionality of iForms **needed the table name of parent forms,** meaning that some functionality **(of the form you are modifying <u>and other forms as well</u>)** will/could break if the table name for the parent form is changed. If you modified a parent form and are satisfied with the changes (and numerous tests), **the final form with modifications has to be renamed to have the original table name.** The backup of the original can have a new name.[75]

3) Make one change at a time and test each change as you do so.

4) By default, iFormbuilder will assign the new copy of the form to the account that made the copy, you might want to un-assign the form if it is sub-form[76] after you are done making changes. If you are modifying a parent form, you have to change the form groups to add the new form and remove the original.

5) If you modified a sub-form and not a parent, you will have to change the link of **"Smart Paging"** tab in the parent form, to link the sub-form element to the new sub-form.

6) Test a hundred more times while enjoying the changes made.

---

[74] You might only want to modify a sub-form or a parent-form. Depending on you needs, you might (or not) need to also make a copy of the sub-forms, if you are making changes to them. If you only are modifying a sub-form, you might just want to copy the sub-form.

[75] One example could be: cd_*2014_cfw_timesheet_master_0214_**original***

[76] Sub-forms should never be assigned to users, only the parent forms. However if you are making a change on a sub form, this helps for testing purposes.

Last reminder

Please erase all test records from the database before taking the live (to collect real field data). But please do create test records and fake data for simulations. The annex includes some barcodes, beneficiary cards, vendor cards, and vouchers to test and/or to learn how to use the forms. Once you erase the tables, these will not work.

Annex

To test the barcodes below, make sure you cover the other ones to avoid scanning the wrong barcode.

1. Beneficiary cards (if you are too close to the bar-codes, they won't scan)

2. Vendor card

3. NFI Vouchers (HEADS bar-codes)

VO-302001 VO-302017 VO-302100 VO-302101 VO-302200 VO-302300

4. $20NFI Vouchers barcodes (sub-vouchers)

VO-302004A-20 VO-302005A-20 VO-302008A-20 VO-302011A-20 VO-302014A-20 VO-302016A-20

5. $10 (4), $5 (2), and $1 (1) NFI Voucher barcodes (sub-vouchers)

VO-302016B-10 VO-302018B-10 VO-302022C-10 VO-302025C-10 VO-302026I-5 VO-302028I-5 VO-302028J-1

6. Dropbox link to zip file with all files referenced here: http://tinyurl.com/ict4drive

   (Includes all bartender files, voucher maker excel files, other guides and partner logos)

7. CRS global to DRC MEAL page (not available to all global accounts):

   http://tinyurl.com/drivedrc (includes same files as above)

8. For the technical people, the simple JavaScript functions to class the different scores are:

```
function bidonscore(bidon) {
if (bidon > 16 && bidon <= 1000) {xbidon=0}
else if (bidon > 13.7 && bidon <= 16) {xbidon=0}
else if (bidon > 11.4 && bidon <= 13.7) {xbidon=0.5}
else if (bidon > 9.1 && bidon <= 11.4) {xbidon=1}
else if (bidon > 6.8 && bidon <= 9.1) {xbidon=1.5}
else if (bidon > 4.5 && bidon <= 6.8) {xbidon=2}
else if (bidon > 3.5 && bidon <= 4.5) {xbidon=2.5}
else if (bidon > 2.5 && bidon <= 3.5) {xbidon=3}
else if (bidon > 1.7 && bidon <= 2.5) {xbidon=3.5}
else if (bidon > 0.8 && bidon <= 1.7) {xbidon=4}
else if (bidon > 0.0 && bidon <= 0.8) {xbidon=4.5}
else if (bidon > -1 && bidon <= 0) {xbidon=5}
return xbidon;
}

function casseroleserolescore(casserole) {
if (casserole > 5 && casserole <= 1000) {xcasserole=0}
else if (casserole > 4.4 && casserole <= 5) {xcasserole=0}
else if (casserole > 3.8 && casserole <= 4.4) {xcasserole=0.5}
else if (casserole > 3.2 && casserole <= 3.8) {xcasserole=1}
else if (casserole > 2.6 && casserole <= 3.2) {xcasserole=1.5}
else if (casserole > 2 && casserole <= 2.6) {xcasserole=2}
else if (casserole > 1.5 && casserole <= 2) {xcasserole=2.5}
else if (casserole > 1 && casserole <= 1.5) {xcasserole=3}
else if (casserole > 0.67 && casserole <= 1) {xcasserole=3.5}
else if (casserole > 0.33 && casserole <= 0.67) {xcasserole=4}
else if (casserole > 0.0 && casserole <= 0.33) {xcasserole=4.5}
else if (casserole > -1 && casserole <= 0) {xcasserole=5}
return xcasserole;
}

function bassinescore(basin) {
if (basin > 4 && basin <= 1000) {xbasin=0}
else if (basin > 3.4 && basin <= 4) {xbasin=0}
else if (basin > 2.8 && basin <= 3.4) {xbasin=0.5}
else if (basin > 2.2 && basin <= 2.8) {xbasin=1}
else if (basin > 1.6 && basin <= 2.2) {xbasin=1.5}
else if (basin > 1 && basin <= 1.6) {xbasin=2}
else if (basin > 0.75 && basin <= 1) {xbasin=2.5}
else if (basin > 0.5 && basin <= 0.75) {xbasin=3}
else if (basin > 0.33 && basin <= 0.5) {xbasin=3.5}
else if (basin > 0.17 && basin <= 0.33) {xbasin=4}
else if (basin > 0.0 && basin <= 0.17) {xbasin=4.5}
else if (basin > -1 && basin <= 0) {xbasin=5}
return xbasin;
}

function couchagescore(bed) {
if (bed > 1 && bed <= 1000) {xbed=0}
else if (bed > 0.88 && bed <= 1) {xbed=0}
```

```
else if (bed > 0.76 && bed <= 0.88) {xbed=0.5}
else if (bed > 0.64 && bed <= 0.76) {xbed=1}
else if (bed > 0.52 && bed <= 0.64) {xbed=1.5}
else if (bed > 0.4 && bed <= 0.52) {xbed=2}
else if (bed > 0.3 && bed <= 0.4) {xbed=2.5}
else if (bed > 0.2 && bed <= 0.3) {xbed=3}
else if (bed > 0.13 && bed <= 0.2) {xbed=3.5}
else if (bed > 0.07 && bed <= 0.13) {xbed=4}
else if (bed > 0.0 && bed <= 0.07) {xbed=4.5}
else if (bed > -1 && bed <= 0) {xbed=5}
return xbed;
}
function outilsscore(tool) {
if (tool > 5 && tool <= 1000) {xtool=0}
else if (tool > 4.6 && tool <= 5) {xtool=0}
else if (tool > 4.2 && tool <= 4.6) {xtool=0.5}
else if (tool > 3.8 && tool <= 4.2) {xtool=1}
else if (tool > 3.4 && tool <= 3.8) {xtool=1.5}
else if (tool > 3 && tool <= 3.4) {xtool=2}
else if (tool > 2 && tool <= 3) {xtool=2.5}
else if (tool > 1 && tool <= 2) {xtool=3}
else if (tool > 0.7 && tool <= 1) {xtool=3.5}
else if (tool > 0.3 && tool <= 0.7) {xtool=4}
else if (tool > 0.0 && tool <= 0.3) {xtool=4.5}
else if (tool > -1 && tool <= 0) {xtool=5}
return xtool;
}

function couverture(sheet) {
if (sheet > 1 && sheet <= 1000) {xsheet=0}
else if (sheet > 0.88 && sheet <= 1) {xsheet=0}
else if (sheet > 0.76 && sheet <= 0.88) {xsheet=0.5}
else if (sheet > 0.64 && sheet <= 0.76) {xsheet=1}
else if (sheet > 0.52 && sheet <= 0.64) {xsheet=1.5}
else if (sheet > 0.4 && sheet <= 0.52) {xsheet=2}
else if (sheet > 0.3 && sheet <= 0.4) {xsheet=2.5}
else if (sheet > 0.2 && sheet <= 0.3) {xsheet=3}
else if (sheet > 0.1333333333333333 && sheet <= 0.2) {xsheet=3.5}
else if (sheet > 0.0666666666666667 && sheet <= 0.1333333333333333) {xsheet=4}
else if (sheet > 0.0 && sheet <= 0.0666666666666667) {xsheet=4.5}
else if (sheet > -1 && sheet <= 0) {xsheet=5}
return xsheet;
}

function habit_femme(woman) {
if (woman > 5 && woman <= 1000) {xwoman=0}
else if (woman > 4.6 && woman <= 5) {xwoman=0}
else if (woman > 4.2 && woman <= 4.6) {xwoman=0.5}
else if (woman > 3.8 && woman <= 4.2) {xwoman=1}
else if (woman > 3.4 && woman <= 3.8) {xwoman=1.5}
else if (woman > 3 && woman <= 3.4) {xwoman=2}
else if (woman > 2.25 && woman <= 3) {xwoman=2.5}
else if (woman > 1.5 && woman <= 2.25) {xwoman=3}
else if (woman > 1 && woman <= 1.5) {xwoman=3.5}
else if (woman > 0.5 && woman <= 1) {xwoman=4}
else if (woman > 0.0 && woman <= 0.5) {xwoman=4.5}
else if (woman > -1 && woman <= 0) {xwoman=5}
return xwoman;
}

function habit_enfant(child) {
if (child > 5 && child <= 1000) {xchild=0}
else if (child > 4.6 && child <= 5) {xchild=0}
else if (child > 4.2 && child <= 4.6) {xchild=0.5}
else if (child > 3.8 && child <= 4.2) {xchild=1}
else if (child > 3.4 && child <= 3.8) {xchild=1.5}
else if (child > 3 && child <= 3.4) {xchild=2}
else if (child > 2.25 && child <= 3) {xchild=2.5}
else if (child > 1.5 && child <= 2.25) {xchild=3}
else if (child > 1 && child <= 1.5) {xchild=3.5}
else if (child > 0.5 && child <= 1) {xchild=4}
else if (child > 0.0 && child <= 0.5) {xchild=4.5}
else if (child > -1 && child <= 0) {xchild=5}
return xchild;
}
```