

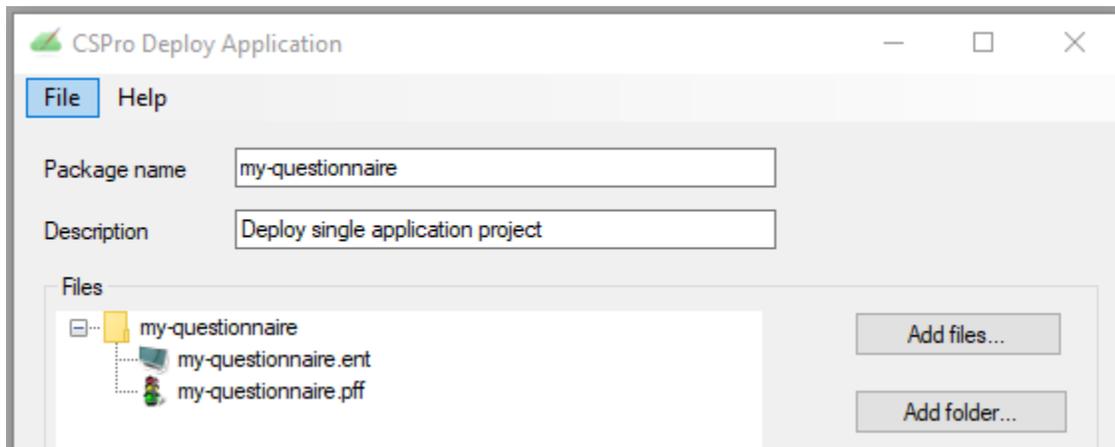
What Type of Server to Use?

If your organization has experience with web server maintenance and cyber security, we recommend using CSWeb as your server. It is optimized to handle small to very large surveys and censuses. However, if your organization lacks the experience to work with CSWeb we recommend using Dropbox as the server. It is appropriate for small to medium size surveys and requires no server setup or maintenance and avoids the cost and difficulty of setting up a CSWeb server. Depending on your organization, a downside to using Dropbox is that your files will reside on their servers. If that is an issue and your survey is small to medium the alternative is setting up an FTP server.

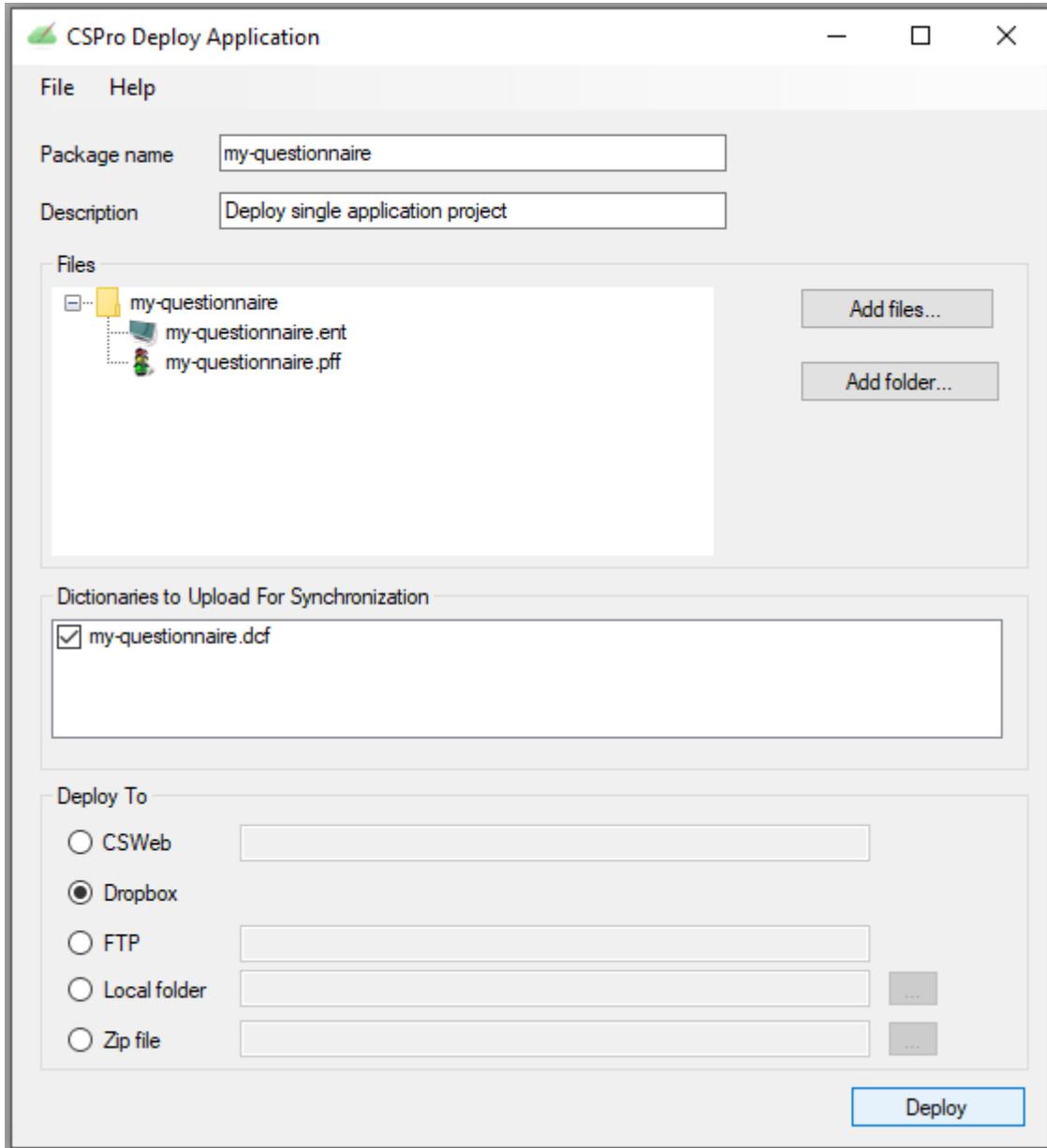
CSPRO Deploy Application

As the application developer CSPRO Deploy Application allows you to bundle up your project and deploy it to a server once, so field staff can download it over the internet at their convenience. Here you will learn how to deploy a single application project.

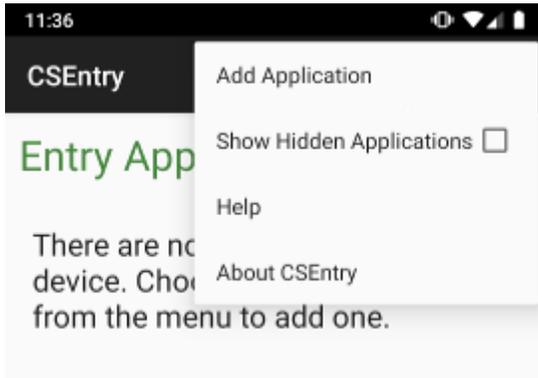
To get started you will need a Dropbox account and a single application project. This document will refer to this application as *my-questionnaire*. Next, open CSPRO Deploy Application which can be found under the *Tools* menu in CSPRO. Use *my-questionnaire* as the *Package name* and "Deploy single application project" as the *Description*. Drag and drop the directory that contains the *my-questionnaire* application onto the white box under *Files*. Note that only the *ent* and *pff* files are added. The *ent* file will be deployed as a *pen* file and any other files that you want to deploy can be added manually by clicking *Add files...*



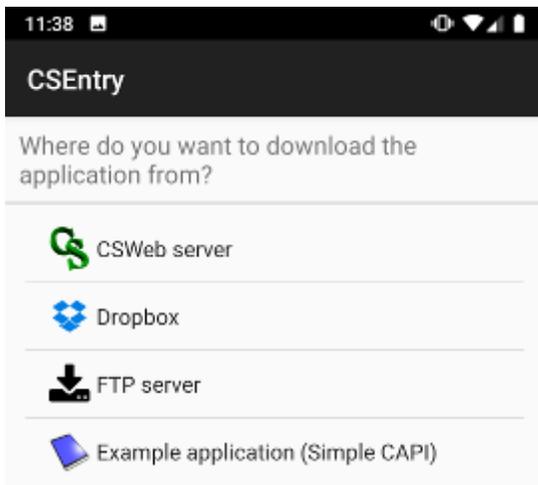
By default, the main dictionary will be checked. This will upload the *my-questionnaire* dictionary to the server, preparing the server, so field staff can upload their collected data. As a final step, select the *Dropbox* radio button and click *Deploy*. If this is the first time deploying to Dropbox, you will be prompted to enter your credentials.



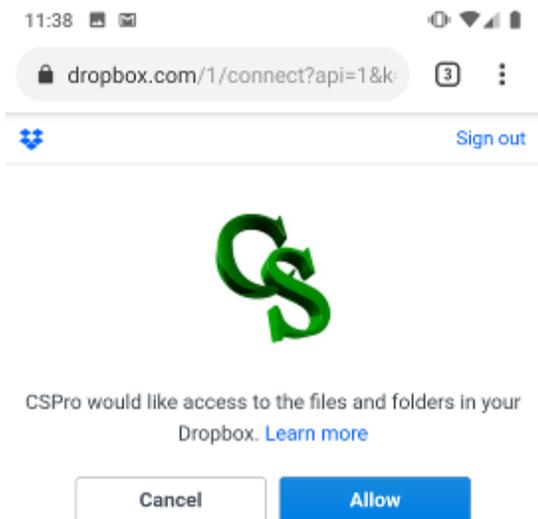
Now field staff, with their device and an installation of CSEntry can download the deployed application by choosing *Add Application* from the CSEntry menu.



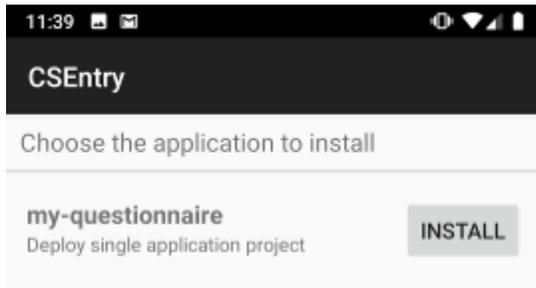
Click *Dropbox* from the list of servers.



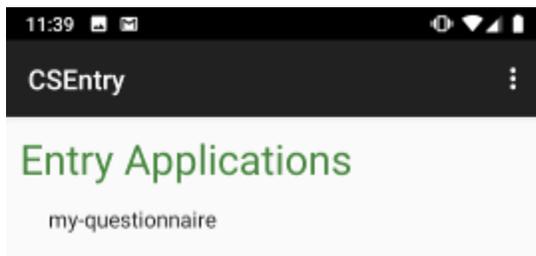
If this is your first time adding an application from Dropbox you will be prompted to enter your credentials and will need to allow CSPro to access your files and folders in your Dropbox.



Click *INSTALL* to download and install the *my-questionnaire* package on the device.



The *my-questionnaire* application will be installed and appear in the *Entry Applications* ready to be launched.



CSPro Synchronization

CSPro has two different methods of implementing synchronization in your project. The first implementation, *Simple Synchronization* is useful when you have only a questionnaire application. It is quick and easy to add to your project, but not as flexible as *Synchronization from Logic*. *Synchronization from Logic* is useful to synchronize multiple applications or additional files or to synchronize from device to device (Bluetooth).

When to use *Simple Synchronization*:

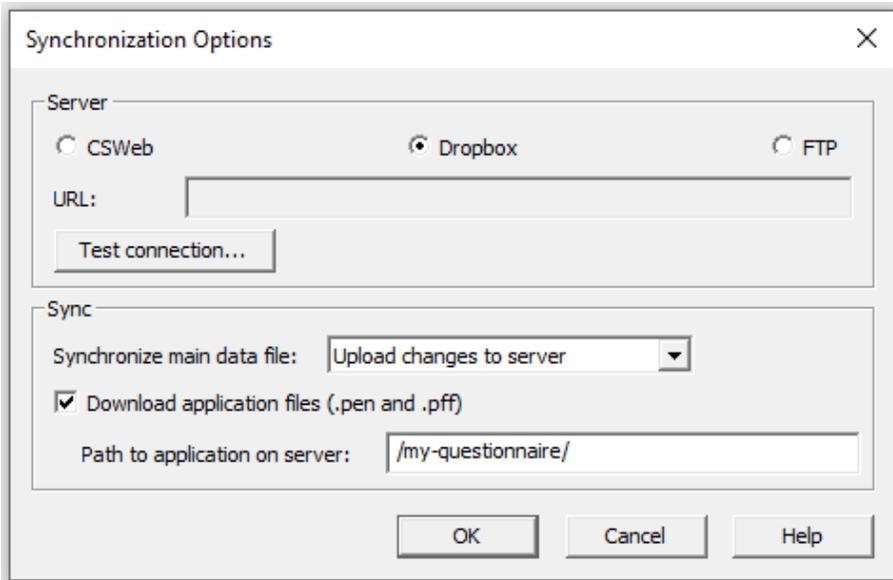
1. Synchronizing the main dictionary (single application project)
2. Downloading application files (*pen* and *pff*)
3. Server is CSWeb, Dropbox, or FTP

When to use *Synchronization from Logic*:

1. Synchronizing external dictionaries (requires a menu application in addition to the questionnaire application)
2. Downloading multiple application files (*pen* and *pff*)
3. Synchronizing additional files
4. Server is CSWeb, Dropbox, FTP, or another device (Bluetooth)

Simple Synchronization

Below are the steps to add *Simple Synchronization* to your single application project. Open the *Synchronization Options* from the *Options* menu.



1. The first option allows you to select the server. If CSWeb or FTP is selected the URL will need to be provided. We recommend Dropbox for the for its simplicity, so select that. The Dropbox URL is already known, so it will be unselectable.
2. The second option specifies whether the collected data will be uploaded, downloaded, or both. For the *my-questionnaire* application upload the enumerator's collected work.
 - a. **Upload changes to server:** Only data that are modified on the device will be sent to the server. No data will be downloaded from the server. This is the most common option for interviewers when all interviewers will work on unique assignments.
 - b. **Download changes from server:** Only receive modified data from the server. Do not upload local changes. This might be used for a supervisor who wants to see what changes interviewers have made but does not want to make changes themselves.
 - c. **Sync local and remote changes:** Send local changes to the server and download changes from the server. This option can be used when multiple interviewers need to work on the same assignments. Note that if both interviewers modify a case at the same time, one will overwrite the changes made by the other.
3. The third option enables or disables the downloading of the application files. If it is checked the path given in *Path to application on server* will be used, otherwise it will be ignored. We will check it and specify the path.
 - a. The path */my-questionnaire/* is specified, so this is where *Simple Synchronization* will expect to download the application files from. Their names must match the name of the application. Since the application is named *my-questionnaire*, the application files must be named *my-questionnaire.pen* and *my-questionnaire.pff*.
 - b. Note that after downloading an updated application, the application must be restarted, otherwise the changes will not be seen.

Synchronization from Logic

Now add a listing application to your project. To synchronize the data and application files for multiple applications you will need to write custom logic. A *Synchronization from Logic* implementation will only synchronize external dictionaries, so you will also need to add a menu application. A basic menu application will launch your applications and synchronize your data and files.

Write your *Synchronization from Logic* implementation, so that it has the same behavior as the *Simple Synchronization* implementation above, but also handle the *my-listing* application. Therefore, you will:

1. Use Dropbox as the server
2. Upload the data to the server
3. Download updated applications

To synchronize with a server, you will need to first connect to it. Do this by calling the *syncconnect* function and providing the server information. Wrap the *syncconnect* function within an *if* statement, so synchronization logic only runs if a connection is established.

```
if syncconnect(Dropbox) then
```

Next, upload the data by calling the *syncdata* function and providing the direction and dictionary name.

```
    syncdata(PUT, MY_LISTING_DICT);
    syncdata(PUT, MY_QUESTIONNAIRE_DICT);
```

To download the application files call *syncfile* and provide a direction, *from path*, and *to path*. A *GET* direction will download the files. The *from path* and *to path* are relative to this direction. With a *GET* direction the *from path* will refer to a source path on the server and the *to path* will refer to a destination path of the device. On the server your application files will reside in the application directory along the path */my-project/*. Since the menu application is currently running, its directory is the current working directory and all of the *to paths* will be relative to it.

```
    syncfile(GET, "/my-project/my-menu/my-menu.pen", "./menu.pen");
    syncfile(GET, "/my-project/my-menu/my-menu.pff", "./menu.pff");
    syncfile(GET, "/my-project/my-listing/my-listing.pen",
        "./my-listing/listing.pen");
    syncfile(GET, "/my-project/my-listing/my-listing.pff",
        "./my-listing/listing.pff");
    syncfile(GET, "/my-project/my-questionnaire/my-questionnaire.pen",
        "./my-questionnaire/my-questionnaire.pen");
    syncfile(GET, "/my-project/my-questionnaire/my-questionnaire.pff",
        "./my-questionnaire/my-questionnaire.pff");
```

Finally, you will call the *syncdisconnect* function to disconnect from the server and end the *if* statement.

```
    syncdisconnect();
endif;
```

CSPro 7.0+ includes an example application called *Synchronization in Logic* that demonstrates the *Synchronization from Logic* implementation.

Bluetooth Synchronization

Synchronizing between two devices (phone or tablet) is similar to synchronizing to Dropbox, FTP, or CSWeb. You can use the above logic with some minor modifications. Instead of connecting to Dropbox you will update the *syncconnect* function with the keyword *Bluetooth*. Since you are running logic to connect to the server, you can think of the device that runs this logic as the client. In this example, you are uploading the data and downloading the application files, so here the client is the enumerator's device. Also, the *from paths* must change, by default this path will be relative to the menu application's directory on the server.

```
if syncconnect(Bluetooth) then
  syncdata(PUT, MY_LISTING_DICT);
  syncdata(PUT, MY_QUESTIONNAIRE_DICT);

  syncfile(GET, "./my-menu.pen", "./menu.pen");
  syncfile(GET, "./my-menu.pff", "./menu.pff");
  syncfile(GET, "../my-listing/my-listing.pen", "../my-listing/listing.pen");
  syncfile(GET, "../my-listing/my-listing.pff", "../my-listing/listing.pff");
  syncfile(GET, "../my-questionnaire/my-questionnaire.pen",
    "../my-questionnaire/my-questionnaire.pen");
  syncfile(GET, "../my-questionnaire/my-questionnaire.pff",
    "../my-questionnaire/my-questionnaire.pff");

  syncdisconnect();
endif;
```

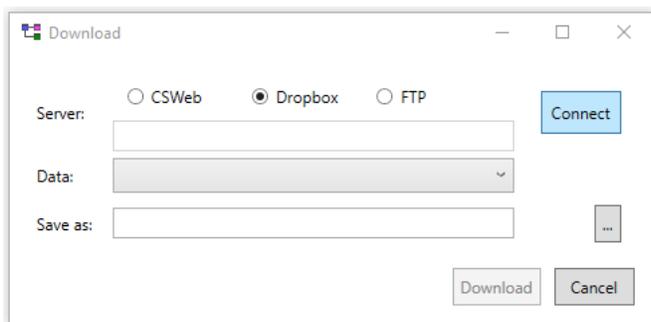
The aspect of Bluetooth synchronization that differs is that a local server must be started on the other device. This device is going to receive the data and have the up-to-date application files, so this server is the supervisor's device. To start the local server, call the *syncserver* function.

```
syncserver(Bluetooth);
```

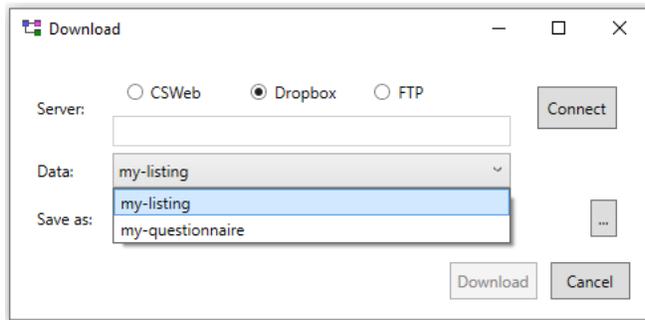
CSPro 7.0+ includes an example application called *Synchronization in Logic* that demonstrates a Bluetooth synchronization between an enumerator and a supervisor.

Downloading Data from the Server

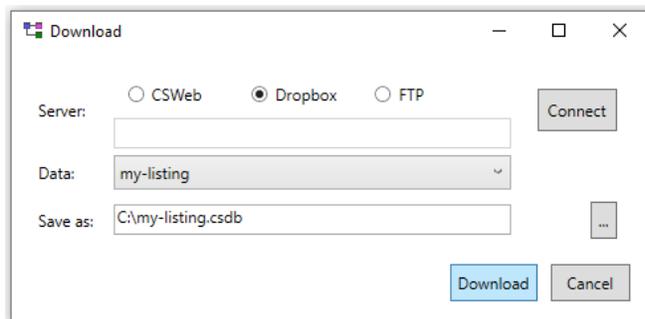
To download the data file (*csdb*) from the server, use the Data Viewer tool. It can be found under *Tools* menu in CSPro. In Data Viewer select *Download...* under the *File* menu. In the *Download* window select *Dropbox* and click *Connect*.



Data Viewer will connect to the server and list the applications with data to download.



Under *Data* select the application, click the ... (browse) button to select where to save the data, enter a filename, and click the *Download* button.



The data for *my-listing.csdb* will be downloaded and opened in Data Viewer.

Dropbox Synchronization FAQ

I do not see my data file in Dropbox?

CSPro's Dropbox sync does not store a single data file in the Dropbox. Instead, it stores a file in */CSPro/DataSync/* for every synchronization. Because of this, you will not see the data as a single file. In order to obtain the combined data file, use the Data Viewer to download the data from Dropbox.

How do I switch to a different Dropbox account?

Windows

1. Go to *Credential Manager*
2. Click *Windows Credentials*
3. Click on *CSPro_sync_Dropbox* and click *Remove*
4. CSPro will ask for the Dropbox credentials when trying to synchronize

Android

1. Hard press on the CSEntry icon
2. Press *App info*
3. Press *Storage*
4. Press *Clear Cache* (this removes the Dropbox information)
5. CSPro will ask for the Dropbox credentials when trying to synchronize