

CommCare Data Export Tool



This feature requires a [CommCare Software Plan](#)

This feature (CommCare Data Export Tool) is only available to CommCare users with a Standard Plan or higher. For more details, see the [CommCare Software Plan page](#).

- [Summary](#)
 - [Exporting to file vs to a database](#)
- [Prerequisites](#)
- [Installing commcare-export](#)
 - [Windows](#)
 - [MacOS](#)
 - [Linux](#)
 - [Upgrade the tool to the newest version](#)
- [Demo data and examples](#)
- [How to Generate an Excel Query File](#)
 - [Excel Worksheet Naming](#)
 - [Columns of a Configuration Sheet](#)
 - [How To Reference Fields in Forms and Cases](#)
 - [Aggregation and Calculation Functions](#)
 - [Example Query File](#)
- [Exporting to SQL](#)
 - [Supported SQL Databases](#)
 - [Data Type Conversion](#)
- [Running commcare-export on the Command Line](#)
 - [Required commcare-export flags](#)
 - [Optional commcare-export flags](#)
- [Using the CommCare Data Export Tool to extract data from a domain with Two Factor enabled](#)
- [Troubleshooting](#)

Latest version

The latest released version of the CommCare Export Tool can be found at:

- [PyPi](#)
- [GitHub](#)

Other useful pages

- [Selecting Tools for CommCare Data Processing and Analysis](#)
- [Tips and Tricks](#)
- [Field reference](#)
- [Advanced features](#)

Summary

`commcare-export` is a command line tool designed to let you export your data from CommCare HQ in a simple, configurable, and reusable way. `commcare-export` will automatically pull your form and case data from CommCare HQ and save it locally, typically to either an Excel spreadsheet or to a local database (like MySQL). You can configure it to pull data on a schedule to automatically keep your data up to date, or run it as a one-time data dump. Here are the basic steps:

1. Install the `commcare-export` tool
2. Create an Excel Query File
3. Write a command-line query to run export
4. (Optionally) Configure export to be re-run automatically
5. Use data

The rest of this page describes how to use `commcare-export` in more detail.

Who should use this tool

- You are looking to export data from CommCareHQ into a SQL database or Excel Workbook.
- You are **moderately technical**, comfortable running command-line programs on your platform of choice.

Exporting to file vs to a database

One major consideration is whether to export to a file (Excel, CSV, JSON, etc) or to a SQL database. Exporting to a file allows you to get a full results set without setting up a local database. However, the major advantage of exporting to a database is that `commcare-export` supports *incremental*/updates. By default, each time the `commcare-export` command is run, it will only import results that were modified after the most recently imported form or case. This means that, for very large datasets, you do not need to export a full results set every time you need to access your data. This can be an incredibly powerful feature for a data pipeline. (See more information about [Selecting Tools for CommCare Data Processing and Analysis](#) here).

Prerequisites



If you are looking to export into a SQL database, you must have this database set up, and you should know how to provide a URL for it.

- **Standard Plan or higher**, your project space must be on the Standard Plan or higher in order to use this tool.
- **Edit Data permissions**, the user that you use to authenticate with must have **Edit Data** permissions.

Installing commcare-export

This section guides you through getting your system ready to use the `commcare-export` command-line tool.

Windows

1. Install Python

Visit the [Python for Windows](#) page and download a package appropriate for your system. The following instructions assume that you have select Python 2.7, but this only affects the location of Python on your hard drive. This tool works with all recent version of Pythons. If you use a different version (i.e. Python 3.2) then you will need to replace each instance of "python27" with your version number (i.e. "python32")

2. Install `pip` (a program that automatically installs Python programs)

Install pip for windows (<https://sites.google.com/site/pydata/log/python/pip-for-windows>)

3. Install `commcare-export`

```
> c:\python27\scripts\pip install commcare-export
```

MacOS

1. Install Python

Some versions of MacOS come with outdated versions of Python. To be safe, you should upgrade; this is very easy. The best way to install software on a Mac is via [Homebrew](#). To install Homebrew and then a modern version of Python, open a terminal and run the following two commands. The first installs Homebrew, interactively telling you each thing it is going to do before it does it. The second installs Python and any prerequisites. (BTW, Mac does not have apt-get)

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"  
brew install python
```

After that, usage of Homebrew is `brew install <SOME PACKAGE>`.

2. Install `commcare-export`

```
pip2 install commcare-export
```

Linux

1. Install Python

If you are on a Linux-based operating system, you probably have an automatic package manager available. Lucky you! You probably already have Python installed or know how to install it, but if not, then you will need to invoke the package manager for your operating system. Here are some possible commands.

```
$ apt-get install python  
or  
$ yum install python  
or  
$ emerge python
```

2. Install `pip` (a program that automatically installs Python programs)

```
$ sudo easy_install pip
```

3. Install `commcare-export`

```
$ sudo pip install commcare-export
```

Upgrade the tool to the newest version

If you have installed an earlier version of the `commcare-export` tool and need to upgrade you can do the following:

Windows

```
> c:\python27\scripts\pip install -U commcare-export
```

MacOS

```
sudo pip2 install -U commcare-export
```



In case you have any problem setting up python on MacOS, reference here: <http://docs.python-guide.org/en/latest/starting/install/osx/>

Linux

```
$ sudo pip install -U commcare-export
```



If you get an error "ValueError: Invalid column index 0" you should run this command to downgrade one of the python libraries:

```
$ pip install -U openpyxl==1.8.6
```

Demo data and examples

This section describes how to run the examples that come with `commcare-export` against a sample project from the CommCareHQ exchange. After working through it, you should have some familiarity with the, its configuration, and how it is used.

1. Set up some demonstration data in CommCareHQ

1. Visit the CommCareHQ Exchange and add the [Simple CommCare Demo/Tutorial](#) app to a new project space.
2. Select the 'Deploy' tab, select 'Make a New Version', star the build.
3. Visit Web Apps and fill out a bunch of forms.

2. Run the examples and see the output in the Terminal

Download the examples [demo-registrations.xlsx](#), [demo-deliveries.xlsx](#) and [demo-pregnancy-cases.xlsx](#) attached to this page. In the following, you will need to adjust the path to the file to be wherever you downloaded them and fill in the name of your project - the bits in bold>. The following commands will run the examples in such a way that the "tables" will be simply printed.

Many different fields are available for output in this way in addition to what is shown in these examples. The fields can be determined from the JSON output as they follow the JSON structure down to each property. For instance, based on the form output here: https://www.commcarehq.org/a/YOUR_PROJECT/api/v0.5/form/ - you would be able to use `form.case`, `@caseid` or `form.meta.username`.

YOUR_PROJECT is the name that you gave to the project space for your application. The name of your project is listed on the first page, Dashboard, when you sign into [CommCareHQ](#). You can also see your project space name in any url while working in your : `www.commcarehq.org/a/yourprojectname`

On Windows:

```
> C:\python27\scripts\commcare-export --output-format markdown --project YOUR_PROJECT --query demo-
registrations.xlsx

> C:\python27\scripts\commcare-export --output-format markdown --project YOUR_PROJECT --query demo-
deliveries.xlsx

> C:\python27\scripts\commcare-export --output-format markdown --project YOUR_PROJECT --query demo-
pregnancy-cases.xlsx
```

On MacOS / Linux:

```
$ commcare-export --output-format markdown --project YOUR_PROJECT --query demo-registrations.xlsx
$ commcare-export --output-format markdown --project YOUR_PROJECT --query demo-deliveries.xlsx

$ commcare-export --output-format markdown --project YOUR_PROJECT --query demo-pregnancy-cases.xlsx
```

Tips:

- 'YOUR_PROJECT' refers to the name of your project space, not the entire domain URL. For instance, the URL would be https://www.comcarehq.org/a/YOUR_PROJECT. For the tool, simply type in YOUR_PROJECT.
- If you are using a domain that isn't stored on www.comcarehq.org, then you will need to use the "--commcare-hq YOUR_URL" flag in the cmd prompt.
- In this example, the output will just be printed in the Command tool. If you want to save the data to a file or database, continue to the next section.

3. Run the examples and save the output to an Excel Workbook

Nearly the same command will save the output to an Excel Workbook.

On Windows:

```
> C:\python27\scripts\commcare-export --output-format xlsx --output registrations-export-results.xlsx --
project YOUR_PROJECT --query demo-registrations.xlsx
```

(try modifying the command above to save the results of each of the examples to a file)

On MacOS / Linux:

(try coming up with this command yourself based on reading the Windows example above!)

Tips:

- If you are using windows, the command line for any path reference to folder names with spaces in the titles need to be included in quotes. For example, the folder 'My Data' would need to be referenced by "C:\Users\YOU\My Data" (quotes included in the command). If you start typing the filepath and when you are almost done, hit the "tab" key, the file name will automatically populate with quotations.
- If you are not going to save data to a SQL data base, you can skip ahead to Excel Configuration Details so learn how to set up the export for your project space.

4. Save the output of an example query to a SQL database



When saving to SQL you should make sure all your sheet names (table names) and field names (column names) are lowercase since not all database correctly handle case.

See [Exporting to SQL](#) section for more details.

By choosing "--output-format sql" and "--output URL" you can save the output to a SQL database. First, let us try it with SQLite, a simple database that is almost certainly already installed and ready to be used on any modern computer, so you do not have to have anything set up to run these examples. In the following, when you see a line that starts with "sqlite>" it is mimicking what it is like to interact with the database, so you should type in the bits that come after that.

In these commands, the term DATABASE_URL is a standard URL format for databases. If, for example, you have a PostgreSQL database hosted on db.example.com named registrations that you would like to save into instead, the URL might look like postgresql://db.example.com/registrations (you may need to include username and password in the URL, but fully explaining the format is beyond the scope of this tutorial).

On Windows:

```
> C:\python27\scripts\commcare-export --output-format sql --output DATABASE_URL --project YOUR_PROJECT --query demo-registrations.xlsx
```

On MacOS / Linux:

```
$ commcare-export --output-format sql --output DATABASE_URL --project YOUR_PROJECT --query demo-registrations.xlsx
```



See [Exporting to SQL](#) section for database URL formats and additional details.

5. Synchronize recent changes periodically

Note: This only applies when exporting to a SQL database.

One expected way to use this tool is to run a weekly/daily/hourly export of just those forms and cases which have changed. To set this up you will need knowledge of how to set up scheduled tasks, so this tutorial will just show a single example command and trust that you are comfortable adjusting it to suit your needs.

When exporting to a SQL database the export tool can pick up where it left off after the previous run. This means that once you have completed an export all you have to do in order to get the most recent data is re-run the same command again.

How to Generate an Excel Query File

This section describes in detail the Excel query file required for commcare-export. You should open one (or many) of the examples ([demo-registrations.xlsx](#), [demo-deliveries.xlsx](#), [demo-pregnancy-cases.xlsx](#)) and compare them to the explanations below.

Excel Worksheet Naming

Each sheet of the configuration workbook is the configuration for an output table. Each sheet should have a unique name. The name of the Excel sheet will become the name of your output table.

The sheet consists of a series of columns with names that are special to commcare-export. These column headers are *required* but also freeing - you may have any other columns you like as long as they do not have one of the reserved titles.

Columns of a Configuration Sheet

- Data Source:** (*required*) Possible values are `form` or `case`.
(*Advanced Usage:* You may specify a deep JSONPath into the data in order to iterate over repeats. For example, if you have a repeat called "child" in the root of the form, the data source would become "form.form.child". Also note that fields are then relative to the source, not absolute. See the [Data Export Tool: Tips and Tricks](#) page for more information. If these terms don't sound familiar, then you probably don't need them!)
- Filter Name:** (*optional*) Including both "Filter Name" and "Filter Value" columns allows you to limit the set of forms or cases that are returned from the commcare-export tool.
 - Forms:** By default, if your data source is `form`, every form submitted to a project space will be returned. Most users will, at a minimum, want to download data for one particular form at a time. To select a particular form type, set Filter Name to `xmlns.exact`

and Filter Value to your form's XML namespace. You can find this for any given form by going to your form -> Advanced -> View (Source XML) and look for the address in the header of your XForm. See additional form filtering options here: [List Forms API](#).

b. **Cases:** If your data source is `case`, every case in a project's `casedb` will be returned. Most users will, at a minimum, want to download data for one `case_type` at a time. To download one particular case type, set Filter Name to `type` and Filter Value to the name of the case type you'd like to download. See additional case filtering fields here: [List Cases API](#).

- Filter Value:** (optional) Filter Names are paired with Filter Values in order to filter the results you request from CommCareHQ. See API documentation listed above for details or acceptable values and formats. Results are limited to the set of forms or cases whose Filter Name property matches the value listed in this field.
- Field:** Each row of this column specifies the name of one `output` field.
Note that for use with a SQL database, a field named `id` is required.
Note that there is no maximum number of output fields.
- Source Field:** Each row of this column specifies where the data of the corresponding **Field** should be extracted from. Note that form properties must be prefixed with `form.` and case properties must be prefixed with `properties.` (See more details about the proper format for Source Fields below in the [How to Reference Fields in Forms and Cases](#) section).
- Alternate Source Field [N]:** (optional) In instances where different source fields should be placed in the same output field (for example when renaming fields in the app) alternate source fields can be specified. Each *Alternate Source Field* column must be named **Alternate Source Field [N]** where "[N]" is replaced by a number. For example:

Source Field	Alternate Source Field 1	Alternate Source Field 2
dob	date_of_birth	date_of_birht

The source fields are each considered in order and the first one that appears in the data is used (even if it's value is empty).

- Map Via:** (optional) This column allows you to convert the raw value to another form (only relevant if you're exporting to SQL). See section [Data Type Conversion Support for SQL](#) below for details
- Table Name:** (optional) By default the table name is taken as the Excel Sheet name. This can be overridden by adding this column with a single value in the column for the table name.
- Include Referenced Items:** (optional) (*advanced*) Each item in this column is a referenced id for which the entire object will be included. For example, when fetching from the case API, if you put `xforms` into this column, then instead of just getting the `xform_ids`, you will get the entire forms in a property called `xforms`. For cases, you can include: `child_cases`, `parent_cases`, `xforms`. For forms, you can include: `cases`.

! Including referenced items in the results may decrease the performance of the tool significantly due to the additional data that is being requested which must be fetched separately from the main form or case data.

Here is an example:

	A	B	C	D	E	F	G
1	Data Source	Filter Name	Filter Value		Field	Source Field	
2	form	xmlns.exact	http://openrosa.org/formdesigner/4EED9EC5-5C14-444F...		id	received_on	
3					Received On	form.name	
4					Woman Name	form.age_calc	
5					Age	form.child_gender	
6					Gender	form.scr_nutrition_grade	
7					Nutrition Grade		
8							
9							
10							
11							
12							
13							

Will be either "form" or "case"

Will be "xmlns.exact" for forms, or "type" for cases (optional)

If filtering by a form, must show the xmlns like the one shown above, or for case type must state the name of the case type (i.e. "patient") (optional)

Get your "Source Fields" from any export- use "form." for form properties and "properties." for case properties. "Field" is the human-readable label.

How To Reference Fields in Forms and Cases

One of the easiest ways to confirm form and case field names is by looking at the column name in a regular form/case export from CommCareHQ.

See also this reference page on commonly exported fields called [Export Field Reference](#)

In general, the name of form and case fields matches those that you wrote when you created your application, with predictable modifications. You should open an example Excel configuration file right now to compare what you see there with the explanations below. When referencing properties in the Source Field column, ensure that you've typed them in exactly as they're defined in CommCare - case sensitivity matters.

Top-Level Form Fields: A field named *FIELD* in a form is referenced by `form.FIELD`. For example, the field `age` is referenced by `form.age` in an Excel config file.

Top-Level Case Fields: A field name *FIELD* in a case is referenced by `properties.FIELD`. For example, the field `delivery_date` is referenced by `properties.delivery_date` in an Excel config file.

Nested Fields: A field nested more deeply within a form or case may be referenced by separating each level of nesting with a period. For example in `demo-deliveries.xlsx` you will see the nested field `form.delivery_information.delivery_type` which will pull out the value the `delivery_type` question inside the group `delivery_information`. This will always match the structure of your application.

(Power Users / Programmers Only) To find out exactly what fields are available, you must view the raw output of the CommCareHQ [Case List API](#) and [Form List API](#). You can then reference these fields using an arbitrary [JSONPath](#) into the structure.

Aggregation and Calculation Functions

The following are functions which can be used to change the value of a column. They can be added to the **Map Via** column in the Excel query.

Function name	Description	Example usage	Since version						
count-selected	Count the number of options selected for a multi-select value.	<table border="1"> <tr> <td>Value</td> <td>a b c d</td> </tr> <tr> <td>Map Via</td> <td>count-selected</td> </tr> <tr> <td>Output</td> <td>4</td> </tr> </table>	Value	a b c d	Map Via	count-selected	Output	4	0.12.4
Value	a b c d								
Map Via	count-selected								
Output	4								
selected-at	Get the option at the given position for a multi-select value. Note: the position index is zero-indexed. This means that to choose the first item, you need to enter 0, the second item, 1, etc.	<table border="1"> <tr> <td>Value</td> <td>red green blue</td> </tr> <tr> <td>Map Via</td> <td>selected-at(2)</td> </tr> <tr> <td>Output</td> <td>blue</td> </tr> </table>	Value	red green blue	Map Via	selected-at(2)	Output	blue	0.12.4
Value	red green blue								
Map Via	selected-at(2)								
Output	blue								
selected	Checks to see if a value was selected from a multi-select value. Outputs 'True' or 'False'.	<table border="1"> <tr> <td>Value</td> <td>rash fever</td> </tr> <tr> <td>Map Via</td> <td>selected(fever)</td> </tr> <tr> <td>Output</td> <td>True</td> </tr> </table>	Value	rash fever	Map Via	selected(fever)	Output	True	0.12.4
Value	rash fever								
Map Via	selected(fever)								
Output	True								
attachment_url	Converts a value into a URL that can be used to download attachments	<table border="1"> <tr> <td>Value</td> <td>photo.jpg</td> </tr> <tr> <td>Map Via</td> <td>attachment_url</td> </tr> <tr> <td>Output</td> <td>https://www.commcarehq.org/a/my-domain/api/form/attachment/the-form-id/photo.jpg</td> </tr> </table>	Value	photo.jpg	Map Via	attachment_url	Output	https://www.commcarehq.org/a/my-domain/api/form/attachment/the-form-id/photo.jpg	0.13.2
Value	photo.jpg								
Map Via	attachment_url								
Output	https://www.commcarehq.org/a/my-domain/api/form/attachment/the-form-id/photo.jpg								

Example Query File

Data Source	Filter Name	Filter Value	Field	Source Field	Map Via
form	xmlns.exact	http://openrosa.org/formdesigner/abcdef	First option	muti_question1	selected-at(0)
			Second option	muti_question1	selected-at(1)
			Num options	muti_question1	count-selected
			Has Fever	muti_question1	selected(fever)

Exporting to SQL



- Note that for use with a SQL database, a field named **id** is **required**.
- Exporting to SQL requires installing additional python libraries. These are listed in the table below.

Supported SQL Databases

Database name	URL format	Maximum field name length	Dependencies
PostgreSQL	<code>postgresql://<username>:<password>@<host>/<database name></code>	63	Requires the 'psycopg2' library: <pre>pip install psycopg2</pre>
MySQL	<code>mysql+pymysql://<username>:<password>@<host>/<database name>?charset=utf8</code>	64	Requires the 'pymysql' library: <pre>pip install pymysql</pre>
MSSQL	<code>mysql+pyodbc://<username>:<password>@<host>/<database name>?driver=ODBC+Driver+17+for+SQL+Server</code>	128	Requires the 'pyodbc' library: <pre>pip install pyodbc</pre> Also requires the SQL Server drivers to be installed.

Data Type Conversion

This is only relevant if you're exporting to SQL.

By default the export tool only supports exporting strings, integer values and boolean values. In order to support dates as well as conversion of other types the following functions can be used in the **Map Via** column:

Function name	Description
<code>str2bool</code>	Convert a string value to a boolean. The following values result in a True output: 'true', 't', '1' (case doesn't matter) If the value is already a boolean then it will be output without change. All other values result in a False output.
<code>bool2int</code>	This supports converting boolean values and booleans represented as strings to the integers 1 or 0.
<code>str2num</code>	Convert a string representation of a number to a number.
<code>str2date</code>	Convert a string representation of a date or timestamp to a date or timestamp.

You can also use the command line option "`--strict-types`" which will prevent SQL column types from being changed. This may result in errors if incompatible data is received from CommCare HQ.

Running commcare-export on the Command Line

Once you have a query file ready, it's time to run your tool on the command line! Here are the required arguments that you will need to provide:

Required commcare-export flags

argument	possible values
<code>--output-format</code>	{json, csv, xls,xlsx, sql, markdown}
<code>--output</code>	Filename or path to SQL database

--project	Project space name (i.e. the PROJECT_NAME value in commcarehq.org/a/PROJECT_NAME/dashboard/project/)
--query	Filepath to Excel query file
--username	E-mail address used for CommCare log-in
--password	Password used for CommCare log-in

Your query should look something like this for:

Excel files

```
$ commcare-export --output-format xlsx --output my_excel_file.xlsx --project PROJECT_NAME --query my_excel_query_file.xlsx --username MY_COMMCARE_LOG_IN_EMAIL --password MY_COMMCARE_PASSWORD
```

SQL imports

```
$ commcare-export --output-format sql --output mysql+pymysql://MYSQL_USER:MYSQL_PASSWORD@127.0.0.1/MYSQL_DB_NAME?charset=utf8 --project PROJECT_NAME --query my_excel_query_file.xlsx --username MY_COMMCARE_LOG_IN_EMAIL --password MY_COMMCARE_PASSWORD
```

Optional commcare-export flags

For a full list of flags, you can run `commcare-export --help`

argument	possible values
--since	DATE value in format YYYY-MM-DD. Will export all data after this date
--until	DATE value in format YYYY-MM-DD. Will export all data before this date
--verbose	Includes more detailed output to the user.
--missing-value	Value to use when a field is missing from the form / case.
--api-version	Specify the version of the CommCare API to use for exports
--start-over	When saving to a SQL database; the default is to pick up since the last success. This disables that.
--strict-types	When saving to a SQL database, don't allow changing column types once they are created.
--version	Output the version of commcare-export that you're using (useful for debugging)

Using the CommCare Data Export Tool to extract data from a domain with Two Factor enabled

If you're using Two Factor Authentication on your domain, you will be required to add an additional argument to the command line prompt: **--auth-mode apikey**. An example query is below.

```
$ commcare-export --output-format xlsx --output my_excel_file.xlsx --project PROJECT_NAME --query my_excel_query_file.xlsx --username MY_COMMCARE_LOG_IN_EMAIL --auth-mode apikey --password API_Key
```

Instead of entering your password, you will be required to enter your API Key as defined under the "My Information" section of "My Account Settings."

Troubleshooting

- **The SQL table or Excel sheet does not get created**
 - The export tool will only create the SQL table or Excel sheet if there is data to export. If there are no forms or cases matching the filters then it will do nothing.
- **The number of results returned does not match what you expect**
 - Make sure you're using the latest version of the CommCare Export Tool (see section: [Upgrade the Tool to the Newest Version](#)).
 - Check the filter fields in your Excel Query file and make sure they are properly formatted.
 - Contact support@dimagi.com if the problem persists.

- **Columns in the Excel Query file are missing from the eventual SQL export**
 - The `commcare-export` tool will only export fields if they exist *in the version of the application that the end user was using when they submitted each particular form and case*. If you are exporting older forms or cases, keep in mind that their property names may be different from what is currently visible in the application.
- **commcare-export seems to be running, but no data is coming in**
 - See note in the **Columns in the Excel Query file are missing from the eventual SQL export** section above.
 - Verify that a `commcare_export_runs` database table has been created, and that the query is visible there.
 - Verify that the sheet name inside your Excel Query file doesn't match another database table that's already been created.
- **commcare-export repeatedly terminates prematurely due to a server error**
 - There are a variety of underlying causes for this. There might be transitory server issues, there might be server maintenance going on, there might be a timeout due to too much data being downloaded.
 - After retrying your query a couple times, you should contact support@dimagi.com directly for help