

**Transcript of:** Data Dictionary

**Video production date:** May 21, 2013

**Video length:** 10:18

**REDCap version featured:** 5.3.4 (standard branch)

**Author:** Veida Elliott, Vanderbilt University Medical Center, Institute for Clinical and Translational Research

*Audio is in normal text below. Timestamps are in bold to assist in finding specific topics.*

**(0:00)**

The data dictionary is a spreadsheet defining the structure of your project. That structure controls the content and functionality of webpages during data entry. There is a one-to-one relationship between the data dictionary and the REDCap project. Each question or field on every data collection instrument occupies a row in the spreadsheet. The order of fields and forms in the dictionary defines the order of their appearance in the REDCap project.

**(0:36)**

The dictionary is just one way to define the data being collected. The spreadsheet is downloaded, modified outside REDCap, and then uploaded back into a REDCap project to change the project structure.

The dictionary is most helpful when using advanced field types such as calculated or matrix fields. It is also the fastest and easiest way to copy several fields at once. The dictionary contains column headers describing what information to include in the cells below. For every data entry field on every row, there are four required columns.

**(1:16)**

First is the variable or field name. Variables names are a keyword or short phrase describing what kind of data will be captured in this field. Names must be unique within the project. They also must use only lower case letters, numerals, and the underscore character.

The variable name is used in saving data, doing exports or reports, and advanced features like calculations or branching logic. The variable name will not be displayed on instruments and is not used during common daily project operations.

**(1:54)**

The form name is the second required column. Forms are the webpages used for data entry. In a project, fields are grouped into forms to organize the data for easier use. Form names should be brief but descriptive. Like variable names, they may contain only lower case letters, numerals, and the underscore character. REDCap will automatically insert capitalization and spacing when the dictionary is uploaded.

The order of form names exactly defines the order of their appearance in the project. Repeat the form name on each row to define which fields are displayed on each form.

**(2:39)**

The third required column is the field type. This column defines the type of question. There are several field types available. To become familiar with the syntax used here, refer to the REDCap “Help & FAQ” tab.

You may find that more than one type of field could be used to capture a given type of data. For example, you may have a survey respondent type in their gender using a text box, or you may give them a multiple choice question to simply choose their answer. The challenge is to determine the field type best suited to your needs.

**(3:21)**

The fourth and final required column is the field label. This column defines the actual questions or descriptive text that will be displayed on the instrument during data entry. The field label column may include any alphanumeric text, including spaces and special characters.

**(3:42)**

These are the four required columns for every field. However, note that an additional column is needed when using certain field types. This column defines the multiple choice answers, calculated equations, or sliders labels used in those field types.

**(4:00)**

When defining multiple choice answers, a unique number must be assigned to each choice. The numbers define how the data will be stored and later retrieved and analyzed. They are not seen on screen in your project. The numbers do not need to be consecutive, and there is no limit to the number of answer choices. Format the answer choices as shown in the demonstration file within any project.

**(4:31)**

When defining equations for calculated fields, only mathematical operators and parentheses can be used. The equation shortcuts used in Microsoft Excel do not translate into REDCap. Equations must be spelled out in their simplest format. Refer to the “Help & FAQ” tab for detailed examples of equation syntax.

**(4:54)**

All other columns in the data dictionary are optional. You do not have to complete every cell for every column. The remaining columns are used to further customize your project and strengthen your data collection strategy.

**(5:09)**

Before exploring the optional columns, there are two final points about variable names. Notice that the first variable name in the data dictionary corresponds to the very first field of the project. This field defines the record names, which are either typed in manually or assigned automatically by REDCap. The first variable name must therefore use a “text” field type. The field label is not restricted in any way. This field is truly just the name or number of the record. It is commonly called patient ID, sample name, or simply record number.

**(5:49)**

The second important point about variable names is that they cannot be changed without data loss. Changing a variable name is essentially deleting it and adding an entirely new one. There is no way to rename a variable and retain any data already saved under that original variable name. For this reason, take care to define variable names while in development mode. Once in production mode and collecting real data, variable names should never be changed. To do so would effectively be deleting them, requiring re-entry of any existing data under the new variable.

**(6:27)**

Returning now to the optional columns. Section headers are defined in a separate column in the dictionary. No data is captured in this field. Section headers are defined slightly differently in the dictionary than in the Online Designer, where the header will be rendered as its own field.

**(6:47)**

Field notes are text displayed underneath any field. These are commonly additional instructions or helpful notes to remember during data entry.

Text fields can be validated to ensure that data is entered in a consistent format. Several validation options are available, and ranges can be particularly useful for numeric, integer, or date validations.

Fields marked as identifiers can only be exported by users having high enough permissions.

**(7:22)**

During data entry, branching logic will hide or reveal questions based on pre-defined criteria. In this example, the logic is defined such that when the variable “sex” has the value “0”, the “given birth” field will appear. This corresponds to a gender of “female”, as defined in the previous field. Branching logic should always be tested in the data collection instruments. Refer to the “Help & FAQ” tab for detailed examples of logic syntax.

**(8:02)**

Making a field required will display a prompt if it is left blank during data entry.

Custom alignment changes the field formatting by moving the data entry field to different places on the screen.

The “question number” column defines numbers displayed before each question on a survey. This column is only applicable for surveys not using branching logic.

**(8:28)**

Lastly, the matrix group name, much like form names, identifies the questions that are displayed in a matrix field.

Remember that the data dictionary is a file entirely outside of REDCap. The file must be uploaded into the project in order to execute the structure in REDCap. Uploading a file is essentially re-creating the entire project structure. All fields in the dictionary will replace all fields in the project. They will not add on to the current structure but will instead totally replace it.

**(9:08)**

After saving the data dictionary, it can be uploaded into REDCap. A summary screen will be displayed. It is vital to review this screen and ensure it makes sense for the changes you were trying to execute. This is your chance to stop and double-check the total number of current and new fields. These totals should make sense for the changes you made in the dictionary. If so, execute them in the project to implement the new project structure.

**(9:47)**

The best way to then test instruments is to enter practice data. It is truly critical to test the instruments during development because iterative changes are made immediately and are less likely to impact real project data.

This is especially true for variable names. Using the data dictionary is an iterative process. With a little practice, you will soon learn how to use this powerful method of project development.