



CIVET
Contentious Incident Variable Entry Template
- - - DRAFT - - - *

Philip A. Schrodt
Parus Analytics
Charlottesville, VA
schrodt735@gmail.com

Version 0.1 : March 17, 2015

*Acknowledgements: The development of CIVET is funded by the U.S. National Science Foundation Office of Multidisciplinary Activities in the Directorate for Social, Behavioral & Economic Sciences, Award 1338470 and the Odum Institute at the University of North Carolina at Chapel Hill with additional assistance from Parus Analytics. This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

1 Introduction

This is the documentation for a prototype of the CIVET ¹—Contentious Incident Variable Entry Template—customizable data entry system. CIVET is being developed by the NSF-sponsored project titled “A Method for Leveraging Public Information Sources for Social Science Research” which is creating tools to improve the efficiency of data generation in the social sciences, with an initial focus on coding event data in the domain of contentious political.

The system you are looking at is a rather ugly prototype and provides a testbed of testing the functionality of some of various elements we expect to have in the final system: it is not intended for actual coding at this point. It will also be evolving as we add additional elements.

CIVET is written using the Flask framework, a Python-based system which is widely available on various cloud platforms. The complete CIVET code will be licensed onto one of the open source licenses, probably the MIT license, and provided on GitHub.

The prototype is currently deployed in the cloud on the “Google App Engine”; we currently expect the operational version will be deployed on a UNC server though the code will be available if you wish to create your own local or cloud versions.

2 Interactive Operating Instructions

At present, the CIVET home page is accessed at the Google-generated URL <http://ace-element-88313.appspot.com/>.

From the home page, click the link in the line *Get started by clicking here*

1. Select a text file to edit: you can use either the pull-down menu or radio boxes, then click the **Edit the file button**.
2. Click one of the text entry boxes will highlight the relevant words in text: For demonstration purposes these are words beginning with the letters 'a', 'c', 'd', 'e' and 's'. The 'tab' key cycles between these options, or an option can be selected using the mouse.
3. When a text entry box is active, the first relevant word in the text is highlighted. The right-arrow key will cycle the highlighted word. To copy a highlighted word into the text box, use the down-arrow key.
4. Text can also be selected using the mouse: To copy the selected text into the text box, use the left-arrow key.
5. Text can also be entered manually.

¹<http://en.wikipedia.org/wiki/Civet>

6. To save a set of coded fields, click one of the buttons along the bottom. At present, all three buttons save; we will be adding “cancel” and “reset” options. The options are:

Return to this case: Save, then return to the same text

Select new case: Save, then return to the same text

Download data: Save, then download data as a .csv filr

7. The “CIVET Download” page provides a text box for a file name, and the **Download file** button downloads the coded data. Use the *Start new data file* link to re-start the coding and the *Continue coding with this file* link to continue adding to the existing records.
 - The .csv file contains the variable names in the first line.
 - If the file name does not end in “.csv”, this will be added.
8. To quit the program, just close the window.²

3 Templates

A template file specifies the individual components of the form: these are the familiar components from web forms but the syntax used to specify these is simpler than what you will find in HTML.

CIVET is simply adding these controls to an HTML `<form>` and, as with all things HTML, most of the placement of the fields is handled by the browser.³ CIVET provides some limited formatting through the insertion of line breaks, and with some experimenting you should be able to keep the form from being too ugly.

Commands vary in length and format and each command ends with a blank line (or, if you prefer, the commands are separated by blank lines.)

At present the program does only a very limited amount of error checking; this will be added in the future.

“#” denotes a comment: anything following a “#” is ignored, so lines beginning with “#” are completely ignored.

²This, it turns out, is a HTML/Javascript security feature which prevents rogue websites from closing windows unless they have created the window.

³Writing in HTML5 and CSS, one can actually exercise a very fine degree of control over the placement, but if you are comfortable with that sort of code, you presumably aren’t using CIVET in the first place. That said, you can see the HTML generated by CIVET by using the *View source* option in your browser, and that could provide a starting point for creating prettier code.

- - - Specifying variables - - -

3.1 Specifying variables to save

This command gives the variables that will be saved; these can be in any order but each of these correspond to a **var-name** somewhere in the form

save: comma-delimited-var-name-list

Example:

save: worldregion,eyewitht, groupname,comments

3.2 constant

Sets variable to a constant; this can be used in a **save:** list.⁴

constant: page-text [varname]

Example:

constant: Data set 0.2 [dataid]

- - - Data Entry Fields - - -

3.3 Items in template specification

The commands involve one or more of the following items:

entry-title : This is the title of data entry field. If this ends with / a line-break (
) is inserted after the text. The titles are escaped.

var-name : The text of the variable name for this field; this will be used in the first line of the .csv output file

comma-delimited-option-list : [select and radio] a list of the items that can be selected, separated by commas. A '*' at the beginning of the item means that it will be initially selected.

page-text : Any text

number : An integer

⁴This isn't particularly useful but we will be adding additional fields for date, time, coder id and the like.

3.4 Checkbox

A simple binary check-box. The value of the variable will be first item in the list when the box is not checked; the second item when the box is checked. The * notation on the second item can be used to specify whether or not the box is initially checked.

```
select: entry-title [var-name]
comma-delimited-option-list
```

Example:

```
select: Eyewitness report? [eyewit]
no,*yes
```

3.5 Select from pull-down menu

Pull-down menus—which are called a “select” in HTML—are specified with the syntax

```
select: entry-title [var-name]
comma-delimited-option-list
```

Example:

```
select: Region [worldregion]
North America, South America, Europe, *Africa, Middle East, South Asia,
East Asia
```

3.6 Radio buttons

A series of radio buttons are specified with the syntax

```
radio: entry-title [var-name]
comma-delimited-option-list
```

The entry / in the option list causes a line-break (
) to be inserted

Example:

```
radio: Region/ [worldregion]
North America, South America, Europe, *Africa, /,Middle East, South
Asia, East Asia
```

3.7 Enter single line of text

This creates a box for a single line of text (HTML `type="text"`). The `width = number` is optional and specifies the size of the text entry box in pixels: the default is `width = 32`

textline: entry-title [var-name] width = number
initial-text

Example:

```
select:  Name of group [groupname]
<enter name>
```

3.8 Enter multiple lines of text

This corresponds to an HTML “TEXTAREA” object. The `rows = number cols = number` is optional and specifies the size of the text entry box: the default is `rows = 4 cols = 80`

textarea: entry-title [var-name] rows = number cols = number
initial-text

Example:

```
textarea:  Comments [comments] rows = 2 cols = 64
-- put any additional comments here --
```

- - - Web Page Formatting - - -

The following options can be used to insert text and line feeds into the web page

3.9 Set page title

Sets the title of the web page.

title: page-text

Example:

```
title:  CIVET-based coding form
```

3.10 Insert text

Adds text to the form: the various options follow the usual HTML formats. In interests of simplicity, text is “escaped” so that special characters are not interpreted as HTML: note that this means that in-line mark-up such as `¡¿`, `¡b¿` and `¡tt¿` will not work, so if you need this use the **html:** command. Also keep in mind that these commands need to be separated by a blank line

h1: page-text
h2: page-text
h3: page-text
h4: page-text
p: page-text

Example:

h1: Primary data set coding form

p:Please enter data in the fields below, and be really, really careful!

3.11 Insert HTML

[This command may or may not be added, as it provides some opportunities for mischief. Stay tuned.]

Adds arbitrary HTML code without escaping.

html: page-text

3.12 Insert a line break

Adds a new line in the form

newline:

3.13 Comments

Any text following “#” is treated as a comment and ignored.

Commands can also be cancelled by adding a “-” in front of the command: this will cancel the entire command, that is, all of the lines associated with the command, not just the first line. For visual symmetry, a “+” in front of the command “activates” it, though the command will also be active without the plus.

4 Features

- We are currently developing this based on a fixed frame that is 960 pixels wide and 700 pixels high: this will fit easily on a 1024x768 screen. For contemporary equipment this is probably quite conservative but it is not unheard of for data labs to have older equipment so we’re going with this at the moment

Known Bug List