



# Cal Poly Drupal Webforms Guide

*For Drupal 7 – Updated 03/08/2016*



# Cal Poly Drupal Webforms Guide

## Table of Contents

<b>Welcome to the Webforms Guide!</b> .....	3	<i>Page Break</i> .....	8
<i>Internal or Authenticated Users</i> .....	3	<i>Number</i> .....	8
<i>External or End Users</i> .....	3	<b>Manage Existing Form Components</b> .....	8
<b>Structure of a Webform</b> .....	3	<i>Edit Existing Components</i> .....	8
<b>Creating a Webform</b> .....	3	<i>Delete Existing Components</i> .....	9
<i>Create the Webform Container</i> .....	3	<i>Clone Existing Components</i> .....	9
<i>Create Form Components</i> .....	3	<b>Configure CAPTCHA on a Webform</b> .....	9
<i>Basic Form Component Settings</i> .....	4	<i>What is CAPTCHA?</i> .....	9
<b>Form Component Types</b> .....	4	<i>Enable CAPTCHA on a Webform</i> .....	9
<i>Fieldset</i> .....	4	<i>Disable CAPTCHA on a Webform</i> .....	9
<i>Textfield</i> .....	5	<b>Email Notifications for Submissions</b> .....	9
<i>Textarea</i> .....	5	<i>Configuring the Email Message</i> .....	10
<i>Select Options</i> .....	5	<b>Webform Submission Settings</b> .....	11
<i>Date</i> .....	6	<b>Conditionals</b> .....	11
<i>Email</i> .....	6	<b>Tokens</b> .....	13
<i>File</i> .....	6	<i>Report Columns</i> .....	14
<i>Grid</i> .....	7	<i>Filters</i> .....	14
<i>Time</i> .....	7	<i>Sort By</i> .....	15
<i>Hidden Component</i> .....	7	<b>Closing Thoughts</b> .....	15
<i>Markup</i> .....	7		



## Welcome to the Webforms Guide!

The following is an in-depth guide to Webforms on Cal Poly Drupal sites. If you're brand new to webforms, we recommend you read the Quick Start Guide available on our website first: [drupal.calpoly.edu](http://drupal.calpoly.edu)

Before we get started, we'll define two terms you'll see used extensively in this guide:

### Internal or Authenticated Users

Users who log into the Drupal site to update content or make other changes.

### External or End Users

Public users. They represent the average site user and are usually your intended audience.

## Structure of a Webform

Webforms are made up of several development layers, each of which allows you to define and shape the webform in different ways.

1. The first layer of the webform is the page that contains the webform. This will appear in the content menu and can't be edited with a New Draft, like a Basic Page. Any content added to this container will precede the Webform and can provide context or instructions.
2. The second layer is the Webform itself. Form elements such as text fields, selection menus, radio buttons, checkboxes, etc, are added and configured here.

- i. Please note that because the Webform is a separate layer, the Publishing Options visible in the first layer do not affect it. There is no draft mode for the Webform layer, and any changes made there go live immediately.

3. Advanced layers to webforms allow for further customization and functionality, and include conditionals, validation, and various settings. We will cover each of these separately within this guide.

## Creating a Webform

### Create the Webform Container

1. To create a new Webform, go to Add Content and select the content type Webform.
2. You will see a page with CKEditor, just like you see when creating a basic page. Content you add here will precede the Webform. This can be left blank, or can be used to offer context/instructions relevant to the Webform.
3. After you have this configured, click Save.

### Create Form Components

Form components, or input fields, must be added after you have created the page itself. For a description of different components, please read Form Component Types.

1. Go to the page you just created and select the Webform tab.
2. You can use the Form Components tab to start adding new input fields.
3. Components are created one at a time by adding a component label, using the dropdown to select the component type, and, if desired, using the checkbox to designate a component as a Required field.

The screenshot shows the 'Form components' tab in the Drupal Webform configuration interface. At the top, there are tabs for 'View published', 'New draft', 'Moderate', 'Webform', 'Results', and 'Track'. Below these are 'Form components', 'Conditionals', 'Form validation', 'E-mails', and 'Form settings'. The main area is titled 'Test Form' and contains a table with columns: LABEL, TYPE, VALUE, REQUIRED, and OPERATIONS. The table lists several components: 'Day of the Week' (Select options), 'On-site assistance' (Grid), 'Text Field' (Fieldset), 'Username' (Textfield), and 'Grid Test' (Grid). The 'Username' component is highlighted with a blue border and has its 'REQUIRED' checkbox checked. Below the table is a form to 'Name the component' with a dropdown menu set to 'Textfield' and an 'Add' button.

LABEL	TYPE	VALUE	REQUIRED	OPERATIONS
Day of the Week	Select options	-	<input type="checkbox"/>	Edit Clone Delete
On-site assistance	Grid	-	<input type="checkbox"/>	Edit Clone Delete
Text Field	Fieldset	-	<input type="checkbox"/>	Edit Clone Delete
Username	Textfield	test	<input checked="" type="checkbox"/>	Edit Clone Delete
Grid Test	Grid	-	<input type="checkbox"/>	Edit Clone Delete
Name the component	Textfield		<input type="checkbox"/>	Add



4. After clicking Add, you'll configure the form component and Save it to commit the new input field to your Webform.

Configuration will depend on the type of component being added, and is described in the next couple sections.

### Basic Form Component Settings

When you configure a form component, the available settings that vary depending on the type of component you are working with. Here are some of the basic settings you will see on most form components:

- Label – this is the user-facing name of the component.
- Field Key – this is a machine-readable version of the label. For some form components, attributes can be added to this key to control the way they appear on the Webform, but the Field Key itself is not user-facing.
- Description – text entered here will appear below the form component in small type to help users understand the purpose or how to use the component.
- Required – when checked, users will only be able to submit their form if they have input a value for this component.
- Default Value – the default value of a component can be set either using static text or a token (for dynamic default values). If a default value is not changed, it will still be recorded when the form is submitted. A field set as Required can be submitted if it contains a default value.
- Placeholder – visible in the field until the user begins entering a value. Is it is not replaced, no value is recorded upon form

submission. A field set as Required cannot be submitted if it contains only placeholder text.

- Disabled – this will make the form component visible but not editable to the user. This is commonly used to set a non-editable default value.
- Private – if enabled, this form component will only be visible to users who are logged in to your site and have sufficient permissions to view Webform results.

## Form Component Types

This section explains the different types of form components that are available and can be added to your Webform. The basic steps necessary to add these form components are the same three steps described in the Create Form Components section, but this section will cover basic configuration for each form component type.

### Fieldset

Fieldsets are containers that you can use to group form elements together. This may be useful to organize logical sections of the form such as contact information. Settings include:

- Collapsible – if turned on, this allows users to collapse the fieldset, hiding the input fields contained within it.
- Collapsed by Default – components within the fieldset will not be visible until the user opens the fieldset.
- Hide Component Title – if you do not want the Fieldset to have its own title, turn this on.

To place components inside of a Fieldset, go to the Form Components tab. Click on the four-way arrow symbol to the left of the component you want and drag it underneath the





fieldset. Then, drag it to the right until it indents, nesting it underneath the Fieldset. Click Save to commit your changes.

*Textfield*

Used to collect or display a small amount of text (usually one line), such as how many attendees are expected. Settings include:

- Max Length – set a maximum number of characters for the field.
- Prefix/Postfix – you can add some text to the left or right (respectively) of your text field. Examples include: \$, #, -, lb, kg, %

*Textarea*

This is very similar to the Textfield component, but allows a bigger area to collect or display a much greater amount of text. Settings include:

- Resizable – determines if the user can make the text area larger or smaller.
- Height – component height can be defined by number of rows.

*Select Options*

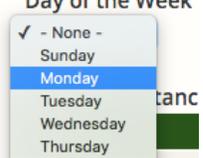
Use this component to create checkboxes, radio buttons, a dropdown, or a list box. Settings include:

- Multiple – if checked, users will be able to select multiple values (checkboxes). If unchecked, users will only be able to select one value (radio buttons).
- Options – this is how you define what options your users have to choose from. The format is “safe\_key|Readable Option,” where “safe\_key” is a machine-readable description of the option, and “Readable Option” is the user-facing name of the option. Place only one option per line.

The “pipe” character (|) is located between the Backspace and Enter keys on most keyboards.

- Load a pre-built option list – allows you to populate the Options setting with a preconstructed list of options such as days of the week. May be useful if you want to use one of these lists, but can also help by providing an example of how to properly format a list of options.
- Listbox – If the component is using radio buttons (see Multiple), this changes the display to a dropdown. If the component is using checkboxes, this changes the display to a listbox.
- Randomize Options – if selected, the options will be presented to the user in a random order.

Visual Samples of Select Options Configurations

<p><b>Day of the Week</b></p> <p><input type="radio"/> Sunday</p> <p><input type="radio"/> Monday</p> <p>Default configuration.</p>	<p><b>Day of the Week</b></p> <p><input type="checkbox"/> Sunday</p> <p><input type="checkbox"/> Monday</p> <p>With multiple.</p>
<p><b>Day of the Week</b></p>  <p>With listbox and multiple.</p>	<p><b>Day of the Week</b></p>  <p>With listbox.</p>

Advanced Select Options Display Settings

Users often format their radio buttons or checkboxes vertically in columns rather than in wide horizontal rows. This is often useful because the rows don’t always format correctly,



especially in the new responsive theme.

To sort options by column instead of row, append “\_cm\_w(X)” to the end of your field key. This breaks down into:

\_cm: means “column major ordering”

\_w(X): means width(X), where X is the number of columns that will display the options.

As an example, if your normal field key is “week\_days,” you can change your field key to “week\_days\_cm\_w2” and get the following formatting:

Day of the Week	
<input type="radio"/> Sunday	<input type="radio"/> Thursday
<input type="radio"/> Monday	<input type="radio"/> Friday
<input type="radio"/> Tuesday	<input type="radio"/> Saturday

### Date

The Date component lets the submitter select a date from drop down lists. Settings include:

- Default value – can be set as a static date (ex, January 1st) or a dynamic date in GNU format (ex, +2 weeks).

More information on the GNU format is available in a link under the Default Value field.

- Default value timezone – determines if the timezone used is based on the user or the site.
- Hide – allows you to hide either the day, month, or year from the user. Any hidden field will be given a default value.
- Start Date – used to set the earliest date the user will be allowed to enter.
- End Date – used to set the latest date the user will be allowed to enter.

- Enable pop-up calendar – enables a Javascript-based pop-up for users to select the date with. Appears next to the date field.
- Use a textfield for year – when checked, uses a textfield for the year instead of the default behavior, which is a select list.

### Email

Used to collect an email address. This is useful because it has email validation built-in, so it requires very little set up and configuration. Settings include:

- User email as default – this won’t work on most Cal Poly Drupal sites. It populates the user’s email automatically, but it can only do so if the user is authenticated (so Drupal knows who they are). End users do not authenticate, so their address is not on file.
- Multiple – allows multiple email addresses if separated by commas.
- Format – can allow either long format or short format.  
Long: “Example Name” <name@adc.com>  
Short: name@adc.com

### File

This component allows users to upload a file. Settings include:

- Upload directory – determines where any files received through this component will be stored in the File Browser.
- Rename files – allows you to create a pattern to be used to rename files that are received. Tokens can be used to create dynamic naming by including things such as the current date.
- Max upload size – used to set the maximum file size that this webform will



accept (up to 50mb but set to 2mb by default).

- Allowed file extensions – determines what type of file will be accepted. Clicking on (select) next to a category such as “Desktop Images” will select all extensions listed under that category. You can also add additional file extensions if something is not listed.

- Questions – define the questions the form will ask your users. These are set up using the same format as the Options.
- Randomize Options/Questions – randomizes the order in which either the Options/Questions appear to the user.

### Grid

Creates a questionnaire-like section with a series of possible answers (options) shared between a group of questions. This is often used for a series of questions answered with a scale like 1-5.

Options appear on the right (and are labeled at the top), while Questions appear on the left.

Settings include:

- Options – this is how you define what options your users have to choose from. The format is “safe\_key | Readable Option,” where “safe\_key” is a machine-readable description of the option, and “Readable Option” is the user-facing name of the option. Place only one option per line.

	1	2	3	4	5
How comfortable is this chair?	<input type="radio"/>				
How awesome is this desk?	<input type="radio"/>				
How pretty is this paint?	<input type="radio"/>				

Adding two readable options, both separated by pipes, allows for semantic differential questions. In other words, to get this:

How much do you like pepperoni?	1	2	3	4	5	
A little	<input type="radio"/>	A lot				

Enter: “safe\_key | A little | A lot”

The “pipe” character (|) is located between the Backspace and Enter keys on most keyboards.

### Time

This component lets users select a time using dropdown boxes. Settings include:

- Default value timezone – determines if the timezone used is based on the user or the site.
- Start Time – used to set the earliest time the user will be allowed to enter.
- End Time – used to set the latest time the user will be allowed to enter.
- Time Format – selects between either a 12 or 24 hour formatting for the time.
- Minute Increments – allows you to set how specific the time entries can be. By default all times can be entered with the 1-minute increments, but settings like 30-minute increments would only allow times on the hour or half-hour to be input.

### Hidden Component

This component is hidden from site visitors and only visible to authenticated users with the necessary permission to access the webform.

There is only one unusual setting in this field:

- Hidden Type – can be set to either “Secure Value,” which allows the use of all tokens, or “Hidden Element,” which is less secure and changeable via Javascript. Neither option is visible to end users.

### Markup



Used to add a section to the form that can be managed using CKEditor. This may be useful if you need to show more information than can comfortably fit inside of a component description, or would like more control over formatting within your form. Settings include:

- CKEditor – you can enter content here just like on a regular page.
- Display On – determines if this field will appear on the form, submission, or both. Submissions are not visible to end users, so setting this to submission only will mean it is only visible to internal users and only after a form submission has been received.

#### Page Break

Allows the webform to be broken up into multiple pages. Inserting page breaks also allows for a “Progress Bar” to become visible. Settings:

- Next/Previous Page Button Label – if left blank, the buttons used to navigate between pages of the webform will be “Next Page” and “Previous Page”. Entering other values in these fields can alter what text appears on those buttons.

#### Number

Collects a numeric value from the user, and has built-in validation options designed for use with numbers. Settings include:

- Integer – if checked, only integer values will be accepted as valid input by the form.
- Minimum/Maximum – defines the lowest or highest numeric value that can be input. Maximum also determines the display width of the field.
- Step – limits options to a specific increment. Ex: a step of 5 only allows

multiples of 5 as input (10, 15, 20, etc).

- Element type – determines if a text field or a select list is used to collect input. A select list may be preferred if validation rules only allow for very specific numbers.
- Prefix/Postfix – you can add some text to the left or right (respectively) of your text field. Examples include: \$, #, -, lb, kg, %
- Decimal places – determine how many decimal places can entered. The default setting (Automatic) allows for 4 decimal places. If the component is collecting a currency amount, this setting will likely use a value of 2.
- Thousands separator – selects whether a comma, period, space, or nothing will separate thousands. Defaults to comma.
- Decimal points – selects whether a comma or period will denote a decimal. Defaults to period.
- Exclude zero – excludes any entry set to 0 or any blank entry when calculating average and standard deviation.

## Manage Existing Form Components

1. Navigate to your webform and select the Webform tab.
2. Select the Form Components tab.
3. Under the Operations column, you can select either Edit, Delete, or Clone.

#### Edit Existing Components

Editing a component brings you to a screen much like what you saw when you first created the component. You can adjust any of the settings you configured when you first added the component.



You can set a component as required on the edit screen, but may find it easier to do so using the checkbox in the “Required” column on the form components tab.

### *Delete Existing Components*

If necessary, you can delete a component, which will completely remove it from the Webform.

Exercise caution when using this option, as component deletion cannot be undone. Not even Drupal Support staff will be able to restore a component if you accidentally delete it.

### *Clone Existing Components*

Cloning a component functions just like adding a new component, but the new component will be a copy of the component you are cloning. It will have the same type, and all of the settings will be configured exactly the same. An Options field in a Select Options component, for example, will be populated with all the same options.

When cloning a component, **be sure you change the Label and the Field Key**. If these are not changed, you may encounter invalid HTML markup errors, and the form may break.

## **Configure CAPTCHA on a Webform**

### *What is CAPTCHA?*

A “CAPTCHA” is a challenge-response test that helps to identify and block spambots from submitting forms. If your webform is receiving a lot of spam submissions, CAPTCHAs can greatly cut down on the number of bogus responses that make it through your form.

### *Enable CAPTCHA on a Webform*

1. Navigate to your webform and select the View Published tab.

2. Scroll to the bottom of the webform and expand the section titled CAPTCHA: no challenge enabled.

3. Click on Place a CAPTCHA here

4. Using the Challenge Type dropdown, select Default Challenge.

5. Click Save. You’ll be taken back to your Webform, and can verify the CAPTCHA has been added if you scroll down and expand the CAPTCHA section again.

- ii. The CAPTCHA section will now read CAPTCHA: challenge “image” enabled

### *Disable CAPTCHA on a Webform*

Disabling a CAPTCHA is done from the same location as it was turned on, at the bottom of the View Published tab of your webform.

Expand the CAPTCHA section and click on Disable. Confirm that you want to disable the CAPTCHA when prompted.

You will be taken back to your form and can confirm that CAPTCHA has been disabled by scrolling down and expanding the CAPTCHA section again.

## **Email Notifications for Submissions**

Drupal Webforms can be configured to send emails when a form submission is received. Emails can be sent as a notification to you or your internal users who need to know when there is a new submission, and they can also be sent to the end user to verify a successful form submission. Many advanced settings are also available.

To add a new email notification, navigate to your webform and select the Webform tab, then the E-mails tab. From here, you can set the



E-mail To using the Address field by listing the email address of the intended recipient. You can list multiple users as well, separating their email with commas. This will start creating an email that sends to the address(es) you listed.

As an alternative to the Address field, the E-mail To value can be set using a Component Value. This can give you a variety of options, depending on the component you select.

Test Form

SEND	E-MAIL TO	SUBJECT	FROM	OPERATIONS
Currently not sending e-mails, add an e-mail recipient below.				
<input checked="" type="checkbox"/>	Address: <input type="text" value="email@example.com"/>			<input type="button" value="Add"/>
<input type="checkbox"/>	Component value: <input type="text" value="Day of the Week"/>			

One of the most common ways to use Component Value is to have an email component collect the user’s email address. This can be used to send a verification email so the user knows their form submission was received.

Other components offer more advanced configuration choices. A Select Options component, for example, will give you an Address field for each Option, and send an email to whichever addresses are associated with the Option the user selects.

To continue this example, if a webform collects requests to reserve a room, but different people are in charge of that room depending on the day of the week, a Select Options component could collect the day of the week and email the appropriate user.

Component e-mail options

OPTION	E-MAIL ADDRESS
Sunday	<input type="text" value="email@example.com"/>
Monday	<input type="text" value="email@example.com"/>
Tuesday	<input type="text" value="email@example.com"/>
Wednesday	<input type="text" value="email@example.com"/>
Thursday	<input type="text" value="email@example.com"/>
Friday	<input type="text" value="email@example.com"/>
Saturday	<input type="text" value="email@example.com"/>

Once you have configured the email to send to the correct addresses, you can begin to build the actual content of the email.

Configuring the Email Message

Enable Sending

This checkbox at the top of the message offers a quick and easy way for you to disable email messages if necessary. This may be useful if you anticipate needing the message to be active again in the future, as it can be used in place of deleting the email and you won’t have to start from scratch to get it set up again.

Email Header Details

Here you can set the subject of the email and define the “from” address and name. If you don’t enter any information here, the webform will try to pull information from your global Webform Settings. If you haven’t entered information there either, the webform will pull from your Site Information. Both Webform Settings and Site Info can be found in the site’s Configuration menu.

These default values may be replaced with a static custom value, or may be based on the value of a component in the webform.

Email Template

This is where the actual content of the email is defined. The default template lists who submitted the form, when they submitted it, the values that each form component collected, and a link that can be used to view the submission (note that this link can only be used by authenticated users, so end users will not be able to see their submission).

The Included E-mail Values section can be expanded to show a list of all components in the



form and a checkbox. Unchecking a component will prevent values it collects from showing up in the email. There is also a checkbox at the bottom that can be used to exclude any “empty” components that were not filled out when the form was submitted.

Customizing the content of the email can be done by adding or removing static text. As an example, adding “Thanks for your submission!” at the end of the template will add exactly that. If you wish to add dynamic content, you will need to use tokens. You can see some of these tokens in use in the default template, and can examine what other tokens are at your disposal by clicking Browse Available Tokens. Some tokens are also discussed later in this guide.

## Webform Submission Settings

A variety of options are available related to form submission, and can be accessed by navigating to the webform, clicking the Webform tab, and then the Form Settings tab.

- Confirmation Message – this CKEditor instance allows you to build a custom message to be displayed when the form has been successfully submitted. Also known as a “thank you” page.
- Redirect Location – determines where the user is sent after they have successfully submitted the form. By default, they are pointed to a page where they see the Confirmation Message, but you can change this to point to a custom URL.
- Status of this form – if set to closed, the form will no longer accept submissions. Closing a form will not break links to it, instead, visitors will see a message displayed to inform them the form is no longer accepting submissions.

- Submission Access – can be used to limit the form such that only authenticated or internal users can submit it.
- Progress Bar – is turned on by default, but only appears on forms that have multiple pages. Can be configured with a variety of information such as whether progress displays by page number or percentage.
- Preview Page – adds a page users can review to double-check the form before submitting.
- Available as Block – enabling this option makes the webform available as a block. This can be useful if you have a form you want available on multiple pages, as all submissions will be consolidated.
- Confidential submissions – when checked, your webform will not collect the IP address a submission originated from. For this to work, the submitter can not be authenticated.
- Submit button label –allows you to override the submission button with a value other than “Submit”.

There are several additional features here that rely on user authentication to function properly. For most Cal Poly sites, your end users will not be authenticated, so these features will not behave as expected, and we advise against using them:

- Total/Per User Submission Limits
- Show “Save Draft” Button
- Show notification about previous submissions

## Conditionals

Conditionals can be used to create advanced webforms that reveal or omit certain questions



based on user input. For example, you could ask users at the beginning of the form to indicate if they are staff or faculty, and use a conditional to show different follow-up questions depending on their response.

It is usually best to configure conditionals after the rest of your Webform. From there:

1. Click on the Webform tab, and then the Conditionals tab.
2. Click on the + icon next to Add a new condition.
3. Use the dropdown menus and text field to configure the conditional.
4. Finally, click Save Conditions.

A conditional will be configured by selecting a series of options that complete an if-then statement, as illustrated below (with the options in brackets):

**If** [form component] [condition] [value],  
**Then** [form component] [condition] [state]

Most of these options are set using dropdown menus, so you don't have to worry about memorizing variables to successfully configure your conditionals.

As an example of a simple conditional, you might set up something like:

**If** [Question A] [is] [Answer X],  
**Then** [Question B] [is] [shown]

You can also nest statements to make compound conditionals. Additional If and additional Then statements can both be added by clicking the + icon that appears next to an existing If or Then.

When multiple If statements are used, an additional choice will appear to select if the statements have an And or an Or relationship. Ex:

**If** [form component] [condition] [value], [**and**]  
**If** [form component] [condition] [value],  
**Then** [form component] [condition] [state]

Selecting the And choice means both If statements must be true for the Then statement to occur. The Or choice means the Then statement will occur when either of the If statements are true.

Multiple Then statements can also be used to affect multiple changes on the Webform based on a single conditional. Ex:

**If** [form component] [condition] [value],  
**Then** [form component] [condition] [state]  
**Then** [form component] [condition] [state]

All of the above options can be combined to create a variety of conditionals ranging from very simple to complex to achieve a variety of tasks. Remember, fieldsets and page breaks are components – so you can reveal or omit multiple components with a single conditional by grouping them in a fieldset or page break.

As an example, consider a form used to submit requests to reserve a room. One question on the form asks the end user if they would like on-site assistance – but the staff don't work weekends, so we can only offer on-site assistance on weekdays.



We built this into the form with the following conditional, which hides the on-site assistance component if the day of the week has been set to Saturday or Sunday. This way, the on-site field is visible until a weekend day is selected.

The above conditional could also be designed to show the on-site question when a weekday is selected. This slight change would mean the on-site field is hidden until a weekday is selected. But, either way – the on-site field is not available when the end user has input a weekend value.

## Tokens

A token is a placeholder for a variable value. The value called by the token depends on what token is being used. Tokens can be used in many different places, ranging from the webform itself to the email notifications it sends, and can accomplish a variety of tasks.

For a full list of tokens, click Browse Available Tokens while working in Drupal. Keep in mind that though many other tokens are available, some of them rely on information that will only be available if the submission came from an authenticated user. Since in most cases end users are not authenticated, some tokens will not function as expected.

Some tokens that will function correctly on most Cal Poly Drupal webforms are listed below:

### Current Date Tokens

These tokens are all based on the current date. Exact display content varies by token.

**Long Format** – [current-date:long]

**Ex:** *Friday, October 9, 2015 - 12:15*

**Medium Format** – [current-date:medium]

**Ex:** *Fri, 10/09/2015 - 12:15*

**Short Format** – [current-date:short]

**Ex:** *10/09/2015 - 12:15*

**Raw Timestamp** – [current-date:raw]

**Ex:** *1444418133*

**Time-since** – [current-date:since]

**Ex:** *45 years 9 months*

### Current Page Tokens

These tokens display a value based on the current page

**Page Number** – [current-page:page-number]

**Title** – [current-page:title]

**URL** – [current-page:url]

### Submission Tokens

Values called by these tokens are related to the webform submissions.

**Date Submitted** – [submission:date]

**Edit URL** – [submissions:edit-url]

**Submission Values** – [submissions:values:?]

The “?” in this token can be replaced with the field key (including any parent field keys separated by colons).



## Webform Reports

Webform Reports are generated by a second module that can tap into the main Webforms module and generate customizable displays of your form submission results. This is done by applying filters to the report that can remove irrelevant submissions or fields.

Reports, like Webforms, are a second layer that live inside of a container (page). And, just like with Webforms, the publishing options only affect the container.

To generate a report:

1. Click Add Content, then Webform Report. As with a Webform, you'll create the container first, and then the actual Report later.
2. Give the report a Title and Section. The optional Description will appear at the top of the Report, if information is entered here.
3. Using the dropdown labeled Webform, designate which Webform your Report will be pulling submission results from.
4. Click Save. Note that reports are not publicly visible, so the publishing options only affects the default view for internal users.
5. Under the Edit tab, select Report Criteria. This takes you to the Report itself.
6. Add columns, filters, and sorts to build your report. Each of these is explained in more detail in the next section.
7. Some additional settings can be adjusted in the Options box at the bottom of the page.
8. Click Update to save your changes.

### Report Columns

Report columns determine what information will be displayed in the report to describe each submission. All form components will appear here, as well as some additional information that may be collected by the form such as the time and date the submission was received. Finally, some functional options also appear that allow the inclusion of links to easily edit, view, or delete a submission.

In the box labeled Add Column, you can select what information you want to add to the report, and then click the Add Column button to insert it. Once inserted, columns can be reordered with the four-way arrow icon (like form components).

Depending on what the information that will be displayed in each column is, some additional options may appear. For example, if a date is added, a dropdown will appear to allow the date display format to be selected – ex: DD/MM/YYYY.

### Filters

Filters can be added almost exactly like Report Columns, but they have an additional dropdown option called Filter Type. This can be set to a variety of qualifiers and is used to include or exclude submissions based on the value of whatever component you have added a filter for.

This is the basic format for a filter:

*[form component] [filter type] [filter value]*

The filter type contains options such as “contains,” “less than,” “is not empty,” and several others. These options, combined with the filter value (when applicable), create the filter.

As an example, say we want to create a report for a room reservation form. Reservations for 50



or more people require additional setup time, so we want to create a report to filter for reservation size to easily identify if any big events are coming up. We would create the following filter:

*[Number of Attendees] [is greater than] [49]*

As with conditionals, there may be multiple ways you could accomplish the same effect using filters. Since the number of attendees should always be a whole number (which I can ensure with form validation if needed), I simply used “is great than 49,” to get the results I wanted.

However, if I was filtering a form component for something like a dollar value of \$50.00 or more, I may have instead used a filter that was set to “greater than or equal to 50,” to maintain accurate results.

Note that all filters are “ANDed” together, which means they will all be applied. So, if we had two filters applied:

*[Number of Attendees] [is greater than] [49]  
[Catering] [equals] [Yes]*

Then the report would show submissions where all applied filters are true. In other words, it only shows reservation requests for 50 or more people that need catering. It would **not** show a reservation request for 55 people if catering was not also requested.

### *Sort By*

The basic interface of the Sort By section works just like the Filters and Report Columns. A dropdown can be used to select the order of sorting that will be applied to the component. This is the basic format:

*[sort attribute] [order]*

Consider the room reservation report that we’ve been using as an example. If we wanted to sort those results by size of attendees, we could use:

*[Number of Attendees] [Descending]*

## **Closing Thoughts**

This covers much of the advanced functionality available in our Drupal webforms. But if you have additional questions, please let us know:

[drupalsupport@calpoly.edu](mailto:drupalsupport@calpoly.edu)

Thanks for reading!