

Quick Guide to Props

Operation and Maintenance Guide

[About this Document](#)

[Introduction to the Quick Guide to Props](#)

[Getting Started](#)

[Prerequisites](#)

[Installation](#)

[Validation](#)

[About the Quick Guide to Ballot Measures](#)

[Data Sources](#)

[Elections and the list of ballot measures](#)

[Text about each ballot measure](#)

[Totals in Money Raised Graph](#)

[Update Frequency](#)

[Largest Contributions](#)

[Update Frequency](#)

[Editing the Quick Guide to Props](#)

[Elections and Lists of ballot measures](#)

[Adding a ballot measure](#)

[Changing the Contents of the Elections Dropdown](#)

[Changing the election for a proposition](#)

[Text data for each ballot measure](#)

[Finding the text for a proposition](#)

[To update an existing proposition](#)

[Adding text for a new proposition](#)

[Totals in the Money Raised Graph](#)

[Largest Contributions](#)

[Adding New Elections](#)

[Troubleshooting](#)

[Pages not displaying / user-visible error messages](#)

[Pages Displaying Incorrect Information](#)

About this Document

This Operation and Maintenance Guide is meant to provide you with all the information that you will need to successfully run the Quick Guide to Props.

This document should serve as a reference to help you understand:

- Where all of the data shown in the program is housed
- How information is imported into the Quick Guide to Props
- How to change text or data that is inaccurate

This document is not meant to provide information on what you can find in the Quick Guide to Props or how to use it - please refer to the User Guide for that information.

Introduction to the Quick Guide to Props

The Quick Guide to Props was developed by MapLight in partnership with the California Secretary of State. This is an open-source project. Code is available in MapLight's GitHub account at <https://github.com/maplight/ca-ballots>

Getting Started

Prerequisites

To host the Quick Guide to Props, you need a running web server with PHP 5.1.6 or newer releases launched by 10/7/2016. Any later PHP updates that deprecate features may not be compatible, and should be tested with the Quick Guide to Props before putting into production use.

Additionally, you need a running instance of [Power Search](#) and access to the tables generated by Power Search.

Installation

1. Install the software listed in prerequisites.
2. Download the Quick Guide to Props. This set of files will be referred to as the project directory.
3. Edit the `application/config.php` file in your project directory.
 - a. Change the `base_url` setting as desired for your environment.

- b. Set the `next_election_date` setting to the date of the election that you want to be displayed by default, in the format YYYY-MM-DD.
 - c. [Optional] Change the log settings and other configurations if desired for your environment.
4. Set up connection to the Power Search Database
5. Copy the contents of the root project directory where it can be accessed by your web server.
6. Deploy the database scripts that will set up the database for the first time.
7. Run database scripts that will populate the guide for each proposition in each election that will be available. See [Adding Text for a New Proposition](#).
8. Adjust the configuration so the program can access the Power Search database and the new tables created in steps #6 and #7.
9. Verify that the amounts in `config/propositions_adjustment.php` are correct for the latest [Political Reform Division analysis](#) of each ballot measure.

As soon as the PHP files are available to your web server, you should be able to begin browsing to the pages. No restart or downtime should be necessary.

Validation

After installation, a number of tests are recommended:

1. Browse to the site root. Verify that all of the elections have been populated. (note: if there is only one election, there will not be a dropdown, and you will be automatically redirected to the page for that election).
2. Browse within each election to verify that the propositions and the information displayed for each ballot measure is correct.

About the Quick Guide to Ballot Measures

Data Sources

This section provides guidance on where the data used by the Quick Guide to Props is stored.

Elections and the list of ballot measures

These lists are generated from the contents of the `caballot_propositions` table.

Text about each ballot measure

Most of the text about ballot measures is copied by hand from the Voter Information Guide into input files. These files are then read by a script and copied into the database, where they are used by the Quick Guide to Props.

The specific regions that are populated in this way include:

- Proposition Number
- Proposition Title
- Summary
- What your vote means
 - Yes
 - No
- Arguments
 - Pro
 - Con
- Campaigns
 - For
 - Against

Totals in Money Raised Graph

Totals in the Money Raised Graph are combinations of data from two sources:

- 1) A query from the tables generated by Power Search, plus
- 2) An offset identified by the Political Reform Division, with amounts for and against each proposition identified in `config/propositions_adjustment.php`. The key used to look up each proposition is the database ID, not the combination of election date and proposition number. For that reason, it is recommended to add a comment specifying the election and proposition number to allow easy maintenance when a new PRD analysis requires these adjustments to be changed.

Update Frequency

These totals will update every time the batch processes for Power Search finish analyzing the latest bulk data from Cal-Access. As of this writing, that update is scheduled daily. The totals will also update instantly whenever you make changes to `config/propositions_adjustment.php` and reload the web page.

Largest Contributions

The numbers shown in the table of largest contributions are drawn from the Power Search database. These are sorted and filtered according to the logic outlined in the User Guide.

Update Frequency

These totals will update every time the batch processes for Power Search finish analyzing the latest bulk data from Cal-Access. As of this writing, that update is scheduled daily.

Editing the Quick Guide to Props

This section provides guidance on how to update or correct information in the Quick Guide to Props.

Elections and Lists of ballot measures

Ballot measures are stored in the ca_search database in the caballot_propositions table.

Adding a ballot measure

Some information about each ballot measure is automatically generated. Specifically, anything stored in the caballot_propositions table is automatically filled in by scripts. These rows should not be edited. The contents are updated daily when the Cal-Access processing scripts are run, and any manual edits will be overwritten at that time.

However, when new ballot measures appear, you will need to also create an entry in config/propositions_adjustment.php with \$0 offsets. See [Totals in the Money Raised Graphs](#) for more information on this.

You will also need to change the specific language for each ballot measure when it is first added. See [Adding Text for a new Proposition](#).

Changing the Contents of the Elections Dropdown

The dropdown menu with a list of elections is also populated from the Election column in this table. To see a list of elections, you can connect to the database and run this query:

```
SELECT distinct Election FROM caballot_propositions;
```

New elections will automatically appear in the dropdown menu when the propositions for that election are posted in Cal-Access.

Text for each ballot measure

Text for each ballot measure is stored in the database in the `cabllot_proposition_languages` table.

The application is currently only available in English.

However, text is stored with a language code as one of the columns to allow easier addition of other languages in the future.

This table contains the following fields:

- **PropositionID** - a reference to the ID for a proposition in the `caballot_propositions` table. Note that this is also the same proposition ID assigned by Cal-Access.
- **Language** - the language code for the text in the field. All entries should be 'EN' for English at this time.
- **Title** - The introductory text provided for each proposition in the Quick Reference Guide section of the official California Voter Guide.
- **Summary** - The summary text provided for the ballot measure in the Quick Reference Guide section of the California Voter Guide. Note that this field can contain limited HTML markup*
- **WhatYourVoteMeansYes** - The text that appears in the Yes column of the What Your Vote Means section. This field can contain limited HTML markup*
- **WhatYourVoteMeansNo** - The text that appears in the No column of the What Your Vote Means section. This field can contain limited HTML markup*
- **ArgumentsPro** - The text that appears in the Pro column of the Arguments section. This field can contain limited HTML markup*
- **ArgumentsCon** - The text that appears in the Con column of the Arguments section. This field can contain limited HTML markup*
- **AdditionalInformationFor** - The text that appears in the For column of the Campaigns section. This field can contain limited HTML markup*
- **AdditionalInformationAgainst** - The text that appears in the Against column of the Campaigns section. This field can contain limited HTML markup*
- **FootnoteYes** - This field is not actively used.
- **FootnoteNo** - This field is not actively used.

*HTML markup in fields is recommended to be limited to simple formatting: bold, italics, line breaks, and similar basic formatting.

If any of the text that is specific to a ballot measure is incorrect, you can fix it by updating the contents of this table. You can also add or remove formatting from the fields that support HTML formatting.

Finding the text for a proposition

To find the text that is being used for any proposition, follow these steps:

1. Find the ID for that proposition in the `caballot_propositions` table. You can do this by looking up the unique combination of ballot number and election date. For example, to find the ID for Proposition 53 in the November 8, 2016 election:

```
SELECT PropositionID FROM caballot_propositions WHERE Number = 53 AND  
Election = '2016-11-08';
```

2. Find the row in the `caballot_proposition_languages` table that corresponds to the proposition ID from step #1 and the language that needs to be changed ('EN'). For example, continuing the above example:

```
SELECT * FROM caballot_proposition_languages WHERE PropositionID =  
1376142 AND Language = 'EN';
```

To update an existing proposition

To update a proposition, first find the row that contains the text you want to change. You can do this using the steps above, in [Finding the text for a proposition](#).

Once you have the PropositionID and Language for the row, prepare a SQL update statement that will modify the column that needs to be changed. Characters within the text that would be interpreted as part of the SQL statement will need to be escaped. Completing this example, let's imagine that we need to add italics to the summary for Prop 53:

```
UPDATE caballot_proposition_languages  
SET Summary = 'Requires statewide voter approval before <span  
style=\"font-style:italic\">any revenue bonds</span> can be issued or  
sold by the state for certain projects if the bond amount exceeds $2  
billion. Fiscal Impact: State and local fiscal effects are unknown and  
would depend on which projects are affected by the measure and what  
actions government agencies and voters take in response to the  
measure's voting requirement.'
```

```
WHERE PropositionID = 1376142 AND Language = 'EN';
```

Adding text for a new proposition

1. To populate text for a whole new measure, first find the PropositionID and Language using the steps in “Finding the text for a proposition”.
2. Verify that there is no existing row already in place. This query, using the correct PropositionID and Language, should not return any rows:

```
SELECT * FROM caballot_proposition_languages WHERE PropositionID =  
1376142 AND Language = 'EN';
```

3. If the above step does return any rows, use the directions [To update an existing proposition](#)
4. If step 2 did not return any rows, create a SQL insert statement that will fill in all of the fields for the new row. Here is an example:

```
INSERT INTO  
`caballot_proposition_languages` (`PropositionID`, `Language`, `Title`,  
`Summary`, `WhatYourVoteMeansYes`, `WhatYourVoteMeansNo`,  
`ArgumentsPro`, `ArgumentsCon`, `AdditionalInformationFor`,  
`AdditionalInformationAgainst`, `FootnoteYes`, `FootnoteNo`)  
VALUES  
(1372638, 'EN', 'Ban on Single-Use Plastic Bags. Referendum.', 'A “Yes”  
vote approves, and a “No” vote rejects, a statute that prohibits  
grocery and other stores from providing customers single-use plastic or  
paper carryout bags but permits sale of recycled paper bags and  
reusable bags. Fiscal Impact: Relatively small fiscal effects on state  
and local governments, including a minor increase in state  
administrative costs and possible minor local government savings from  
reduced litter and waste management costs.', 'A YES vote on this  
measure means: Most grocery stores, convenience stores, large  
pharmacies, and liquor stores would be prohibited from providing  
single-use plastic carryout bags. Stores generally would be required to  
charge at least 10 cents for any other carryout bag provided to  
customers at checkout. Stores would keep the resulting revenue for  
specified purposes.', 'A NO vote on this measure means: Stores could  
continue to provide single-use plastic carryout bags and other bags  
free of charge unless a local law restricts the use of such bags.',  
'YES on 67 protects California’s successful efforts to PHASE OUT  
PLASTIC GROCERY BAGS. Plastic bags strangle wildlife, litter  
communities, raise clean-up costs, clog recycling machines. Bans on
```


plastic grocery bags are WORKING IN 150 CALIFORNIA COMMUNITIES. Don't let out-of-state plastic companies stop California. YES on 67.', 'DON'T BE FOOLED. Prop. 67 is a \$300 million annual HIDDEN TAX on consumers who will be forced to pay \$.10 for every grocery bag at checkout. Not one penny goes to the environment. All \$300 million goes to grocer profits. Stop the bag tax . . . VOTE NO ON PROP. 67.', 'Mark Murray

\nCalifornia vs Big Plastic
\n921 11th Street, Ste. 420

\nSacramento, CA 95814
\n(916) 443-5422

\nmurray@cawrecycles.org
\nprotectplasticbagban.org', 'No on 67

\n2350 Kerner Blvd., Suite 250
\nSan Rafael, CA 94901',
'Committees for Yes on 67
began the election cycle with an <span style=\"font-weight:
bold;\">additional \$125,000 raised earlier.', 'Committees for
No on 67 began the election
cycle with an additional \$3.26
million raised earlier.');

Totals in the Money Raised Graph

Totals from the money raised graph are combinations of a database query in Power Search plus an offset in the configuration file config/propositions_adjustment.php.

If this number is not accurate, it is most likely that the adjustments for the proposition need to be updated. Look at the number for the proposition in config/propositions_adjustment.php. The comments should help guide you, as in this example:

```

88      // Election: 2016-11-08      Prop Num: 65
89      '1381520' => array(
90          'yes' => '3257500',
91          'no' => '0'
92      ),

```

The amount on the 'yes' line is added to the number from the Power Search query of contributions made in support of the proposition, and the amount on the 'no' line is added to the number in the Power Search query of contributions made in opposition to the proposition.

These entries must be **whole dollar amounts** with **no text formatting**. They may be negative, which will cause the Quick Guide to Ballot Measures to subtract, rather than add, the adjustment from the results provided by Power Search.

In the example above, \$3,257,500 is added to the Power Search summary for Prop 65 in the November 8, 2016 election. No change is made to the Power Search summary against Prop 65.

Changing the adjustment in `config/propositions_adjustment.php` will immediately update results for visitors as soon as they reload the page.

Largest Contributions

The table of largest contributions is pulled directly from a query of Power Search data. If you have reason to believe this table is displaying incorrect results, first check the results against the criteria in the User Guide.

If the results still appear wrong, it may help to compare the largest contributions to the results for the equivalent search through Power Search.

If Power Search is also not displaying the correct information, you may need to troubleshoot Power Search's batch processing scripts.

If Power Search is displaying the data correctly but the Quick Guide to Props is not, then you may need to analyze the code for the Quick Guide to Props and identify where the query or display is going wrong.

Adding New Elections

Some information for new elections will be automatically populated in the Quick Guide to Props as the ballot measures for that election appear in Cal-Access. Specifically, the contents of the `caballot_propositions` table are filled in.

However, some information will still need to be added by hand for each proposition:

1. Language about each proposition - see [Adding text for a new proposition](#)
2. An entry should be added to the `config/propositions_adjustment.php` file with \$0 offsets. See [Totals in the Money Raised Graphs](#) for more information about this file.

Troubleshooting

Pages not displaying / user-visible error messages

If the pages are not displaying, check the web server logs. Any errors encountered while serving the PHP pages would likely be recorded in the web server log.

Pages Displaying Incorrect Information

If pages are displaying the wrong information, it is most likely caused by the data being incorrect in one of its sources. Depending on which section of the application, this may be the `caballot_propositions` table, the `caballot_proposition_languages` table, `config/propositions_adjustment.php`, or the tables used by Power Search. See the Data Sources section of this guide to learn where the data for each part of the application is housed and how to modify it.