

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/242456441>

# Guidelines for Creating a Deck of Test Ballots

Article

---

CITATION

1

READS

140

1 author:



[John Washburn](#)

Washburn Research

6 PUBLICATIONS 6 CITATIONS

SEE PROFILE

# Guidelines for Creating a Deck of Test Ballots

By

John Washburn

Certified Software Quality Engineer

1998 - 2006

## Revision and Authorship Notes

Revision	Author(s)	Description	Date
1.0	<a href="#">John Washburn</a>	Initial Revision	March 10, 2005
1.1	<a href="#">John Washburn</a>	Create document which is not Wisconsin specific and which is not specific to any particular voting machinery.	April 1, 2005
1.2	<a href="#">John Washburn</a>	Complete re-write for the new, 5 simple rules approach.	August 20, 2005
1.3	<a href="#">John Washburn</a>	Correct many typographical errors	October 12, 2005
1.4	<a href="#">John Washburn</a>	More corrections of typographical and grammatical errors	March 15, 2006
2.0	<a href="#">John Washburn</a> <a href="#">VotersUnite!</a>	Removed nearly all reference to software testing and refocused the document on the intended audience: election officials and election workers. This simplification results in condensing the document to 2/3 <sup>rd</sup> s of its prior size. The 5 sub-decks were renamed to more election-related names.	September 8, 2006

## Table of Contents

<b>Introduction</b> .....	<b>3</b>
<b>Structure of the Guidelines</b> .....	<b>5</b>
Why so many test ballots? .....	5
Timeline for Tasks in Pre-Election Testing.....	6
Create Your Own Test Deck! .....	7
Perform Tests in “Election Mode” .....	8
<b>Creating a Test Deck Script for Each Ballot Style</b> .....	<b>9</b>
Sub-Deck 1: Straight-Party Test Deck .....	10
Sub-Deck 2: Position Test Deck.....	12
Sub-Deck 3: Over-vote Test Ballots .....	14
Sub-Deck 4: Multi-Vote Test Ballots.....	15
Sub-Deck 5: Tally Distinction Test Ballots.....	17
<b>Using the Test Ballots</b> .....	<b>20</b>
Executing the Test.....	20
For DREs or Other Electronic Devices .....	20
For Units Intended for Use by People with Disabilities .....	21
For Foreign Language Units.....	21
Recording Test Results .....	22
Checking the Accuracy of the Results.....	23
Determining the Cause of Failures .....	24
<b>Appendix A. Creating Scripts for Multi-Mark Test Ballots</b> .....	<b>25</b>
Constructing a Script for Over-vote Test Ballots .....	25
Constructing a Script for Multi-vote Test Ballots .....	26
Generating a Script with 3-Mark Ballots.....	28
Generating a Script with 4-Mark Ballots.....	30
Generating a Script with M-Mark Ballots.....	30
<b>Appendix B. Reporting Defects Found in these Guidelines</b> .....	<b>31</b>
<b>Appendix C: Sample Ballot</b> .....	<b>32</b>
<b>Appendix D: Sample Form – Expected Results</b> .....	<b>34</b>
<b>Appendix E: Sample Form – Test Result</b> .....	<b>35</b>
<b>Appendix F: Where To Find This And Other Documents</b> .....	<b>36</b>

## Introduction

The question to ask when testing voting equipment is:

Does the equipment (hardware and software) read and tabulate the marks on a ballot or touches on the screen with 100% accuracy?

Creating and using the sub-decks described in this document would answer this question completely with a definite yes or a definite no. Moreover, if the answer is *NO*, the information gathered will inform you where the software is inaccurate.

This document was initially intended to help Wisconsin election officials test the software of voting machines for correctness and accuracy. Since then the scope and audience of this document has increased to include other states. This document is intended for people with an interest or statutory duty to conduct testing on voting equipment, such as:

- Programmers, designers or vendors of voting machinery,
- State certification boards,
- Local election administrators such as county or municipal clerks, county or municipal boards of election or registrars of voters
- Independent testing labs hired to perform the testing needed for certification, and, recently,
- Citizen activists interested in election integrity.

Throughout this document, to increase clarity, examples are drawn from a single sample ballot. The specific ballot set up for this document is based on the ballot for District #1, of the Village of Germantown Wisconsin, for the Presidential election held on November 2, 2004. This ballot was chosen for its unique variety; it contains party preferences, more than two parties to prefer, contested races, uncontested races, and ballot questions. The type of contest that was missing from the official ballot is an election for an office for which a voter can cast two or more votes. A contest in which the voter could vote for 3 choices in a set of 6 candidates was added to the sample ballot. For simplicity, some contests were changed to propositions. A copy of this sample ballot is included in Appendix C.

The test ballots created by following these guidelines are designed to fully test the software – NOT to simulate an election. Therefore, the test deck will not resemble a set of ballots cast on election day. However, it will accomplish its purpose, which is to identify errors in the software that would assign votes to the wrong candidate or the wrong response to a question.


Most software problems occur in handling unusual ballots, so testing for the *unusual* is just as important as testing for the usual. Generally speaking, each ballot in the test deck is intended to uncover a single kind of software defect. Thus, any races or ballot lines not required to uncover this defect are left blank by design. For this reason, the deck will contain undervotes in much higher numbers than are normally seen in a real election.

Another reason the test deck will not resemble a real election is that it is intended to check every “if/then” decision required of the software. Voting software is composed of a series of “if/then” programming decisions, for example:


- If there is a mark (screen touch), then...[do this]
- If there is a mark (screen touch) not in the designated area, then...[do this]
- If there is an over-vote, then...[do this]
- If there is a mark for party preference and a write-in, then...[do this]
- If there is a mark for a candidate and a write-in for the same office, then...[do this]

The software must tell the equipment what to do for every possible mark on the ballot or touch on the screen, and every possible combination. Since an error at a decision point will cause the software to make the same mistake every time it makes that decision, it is crucial to test all these possibilities to ensure that the equipment takes the appropriate action for every one of them. This is what the testing guidelines are designed to accomplish – creating a test deck that will make sure the equipment takes the correct action at every “if/then” decision point.


For example, suppose a ballot has two marks for a single office, when the voter was only allowed to vote for one candidate. The software should reject the ballot as an over-vote. Software instructions like the following would accomplish this correctly:

- 1 For the Governor’s contest
  - 2 Examine the ballot
  - 3 If the number of marks is greater than 1, then
  - 4 Reject the ballot
  - 5 Continue with the next ballot until all ballots are examined
  - 6 Go on to the next contest
- 

However, if two instruction lines (4 and 5) were reversed, the last ballot examined would be rejected whether it was over-voted or not, and no other ballot would be rejected.

- 1 For the Governor’s contest
  - 2 Examine the ballot
  - 3 If the number of marks is greater than 1, then
  - 4 **Continue with the next ballot until all ballots are examined**
  - 5 **Reject the ballot**
  - 6 Go on to the next contest
- 

Or if line 3 had an error and checked for more than “0” marks instead of more than “1”, then all marked ballots would be rejected as over-votes.

- 1 For the Governor’s contest
  - 2 Examine the ballot
  - 3 If the number of marks is **greater than 0**, then
  - 4 Reject the ballot
  - 5 Continue with the next ballot until all ballots are examined
  - 6 Go on to the next contest
- 

## Structure of the Guidelines

This document is a set of guidelines. The guidelines are divided into two major sections.

1. Creating the five sub-decks of test ballots in a way that ensures a unique total for each ballot line and calculating the expected vote tallies.
2. Executing the test of the software on the voting machinery.
3. Tracking results and comparing them to the expected results.

For an optical scanning system, these guidelines are used to test the accuracy of the central scanner and the precinct scanners. For a DRE system, these guidelines test the accuracy of DRE itself. These guidelines are designed to uncover defects which are the result of programming mistakes and programmer confusion. They **are not** designed to detect malicious code, such as “ ignore every 20<sup>th</sup> vote for Candidate B.”

DRE equipment is particularly difficult to test well. This is because, for every ballot, you must access the blank ballot, touch the screen, navigate the screens, enter the votes defined for that specific ballot in the deck, then hit the vote button. If the results are not what you expected, you may not know if you made a mistake entering the ballot or if the machine was programmed incorrectly. For this reason, you should video tape the casting of the ballots on DREs.

For optical scanning systems, the test ballots can be created once and reused on as many machines as you need to test. If an error is found, the ballot programming can be corrected and you can use the same ballots to retest the machines.

### **A complete test deck must be created for each ballot style.**

If ballot rotation is used, then there is a separate ballot style for each precinct in the rotation. If candidates for primary ballots are printed on separate ballots for each party, then there is a distinct ballot style for each partisan primary ballot. If the number of races, number of candidates, or the order of candidate names varies from one ballot to another, then there is a distinct ballot style for every unique combination of races, candidates, and candidate listing.

## ***Why so many test ballots?***

The short answer is: Software is difficult to test; especially software with many decision points. The software in voting machinery is nothing *but* decision points;

- Is that a mark or a screen touch?
- Accept the ballot, or reject the ballot?
- Add one to the count of ballots cast?
- Tally a vote for Yes, or tally a vote for No?
- Tally a vote for candidate Alice, or tally a vote for candidate Bob?
- Add one to the count of ballots cast for Party 1?
- Segregate ballots with valid write-in votes?

## Timeline for Tasks in Pre-Election Testing

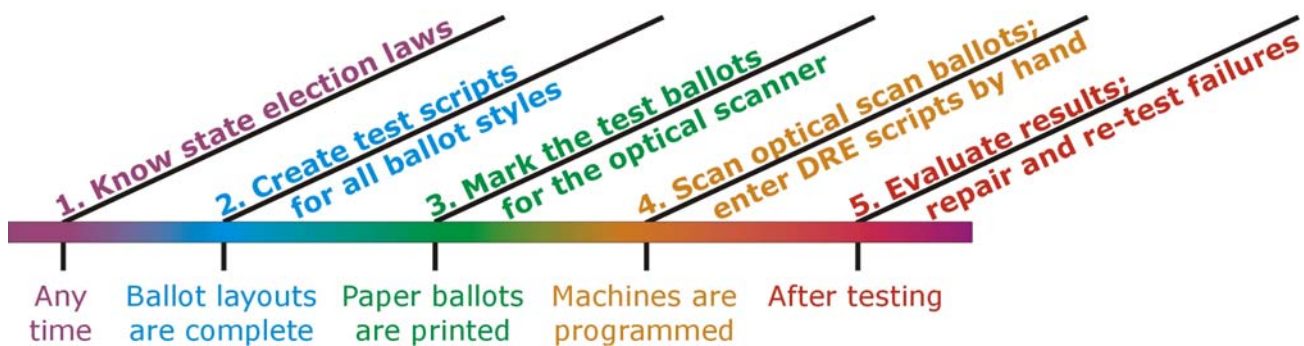
Since the weeks before an election are filled with necessary tasks, it is important to know when you can begin each of the tasks required for adequate pre-election testing. This section gives some guidelines.

- 1) **First** – know your state’s laws and administrative rules about how votes are to be counted. You can do this at any point, since it is not dependent on the details of the ballot styles. For example, how should each of the following ballots be counted? Or should it be rejected?
  - a. Marks for both a straight-party vote and a candidate from a different party in a specific contest.
  - b. Marks for both a straight-party vote and a candidate from the same party in a specific contest.
  - c. Marks for both a straight-party vote and a write-in line for a contest.
  - d. Marks for both a listed candidate and a write-in candidate in the same contest.
  - e. Marks for both a candidate and “None of the Above” (Nevada Only).

The correct behavior of the software for your state depends on the specific state laws and administrative rules. For example, depending on the state, the ballot described in paragraph **d** is:

- Counted as a vote for the written-in candidate,
- Counted as a vote for the pre-printed candidate,
- Or rejected as an over-vote.

- 2) **Second** – create the scripts for the test deck of every ballot style in the jurisdiction as soon as the ballot layouts are finalized. There is no need to wait until you have printed ballots or programmed machines.
- 3) **Third** – if you are testing optical scanning machines, mark the test ballots according to the script as soon as your printed ballot are available. Complete the forms describing the expected results (Appendix D) and create the forms on which you will record the test results (Appendix E).
- 4) **Fourth** – you can begin testing as soon as the machines are programmed with the correct ballot definition information.
  - For optical scanners, this entails running the test deck through the scanner.
  - For DREs, this entails entering each test ballot in the script by hand.
- 5) **Fifth** – compare expected results to actual results. If they don’t match, the machine failed; repair the problem and re-test.



## ***Create Your Own Test Deck!***

This is the first and most important guideline.

Pre-election testing conducted on a touch screen or optical scanning machine is testing software.

The test deck provides the **only** testing that will ever be conducted on the ballot programming before it is used to determine the results for this specific election. The test deck is the only thing that protects the election results from an error in the ballot programming.

Therefore, the test deck must be created by someone who was not involved in creating the program being tested – for two reasons. First, the testing must be done by someone who **wants** to find errors, rather than someone who has a personal investment in finding that the program is error-free. Second, people do not proofread their own documents reliably. Similarly, people don't critique their own work objectively. If we miss an error in the first place, we're likely to miss it again and again.

So also, it is important for someone **other than the person who programmed the machine** to create the set of ballots that tests the programming.

Do not use a test deck provided by:

- The vendor of the voting machine or any employee or agent of the vendor.
- Any programmer who developed the voting machine software, or the employer, an employee, or an agent of any programmer.
- Any person who encoded the ballot structure for the election, or the employer, an employee, or an agent of any person who encoded the ballot structure.

It is crucial for you to create your own deck of test ballots if:

- You are responsible in any way for the correctness of an election tally
- You must certify the correctness of an election tally
- You must certify voting software as correct. Correct here means the software conforms to either statutory requirements or conforms to contractual requirements.

***Create your own test deck!***



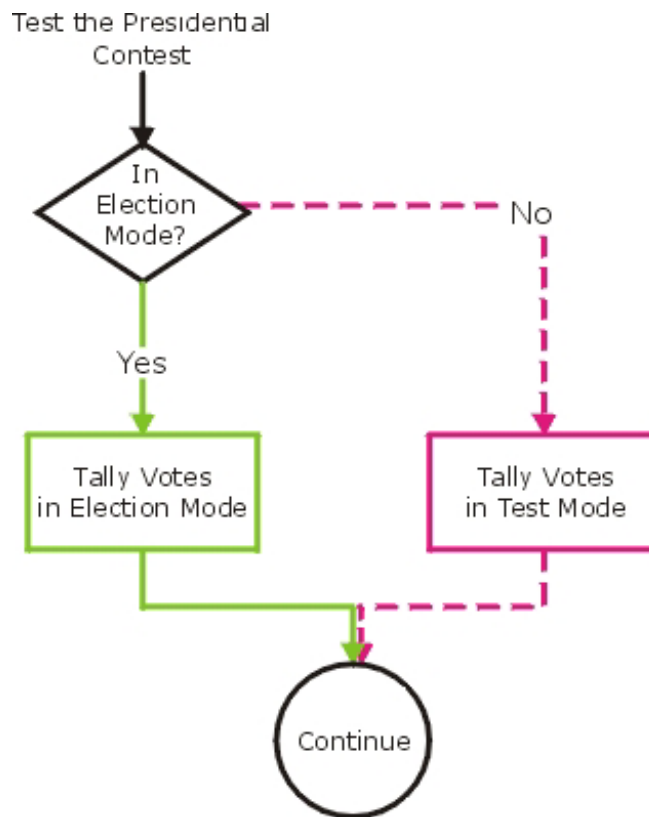
## Perform Tests in “Election Mode”

This is the second guideline.

Many voting machine vendors offer the option of “test mode”, “automatic testing”, “automated testing” or “self testing”. **NEVER** use these features if you must personally attest to the correctness of the software.

If you run the tests in “test mode” or any automatic testing mode, you are NOT testing the software that will be used in the election. You can only attest to the accuracy of the software used in the election if you test THAT software. This means testing in “election mode” only.

Here is a pictorial example.



The “election mode” is the solid, green path. The “test mode” is the dashed, magenta path. The two paths are unconnected. If you test your test ballots on the dashed, magenta path you cannot be certain of the correctness of the solid, green path. Testing one path gives you no information about the correctness (or deficiencies) of the other path.

***Always test the software of voting machinery in “election mode”***

## Creating a Test Deck Script for Each Ballot Style

Create a separate test script for **every ballot style**. You will use this script to create paper ballots or enter ballots on an electronic machine.

A complete test deck is made up of five smaller test sub-decks. Each sub-deck is designed to uncover a specific kind of programming error. This section contains details for creating each sub-deck.

Follow the guidelines in these subsections to create each script. Always create sub-decks 2, 3, and 5. Create sub-deck 1 only if your ballots **include** a straight-party vote. Create sub-deck 4 only if your ballots include one or more contests in which a voter can vote for more than one choice.

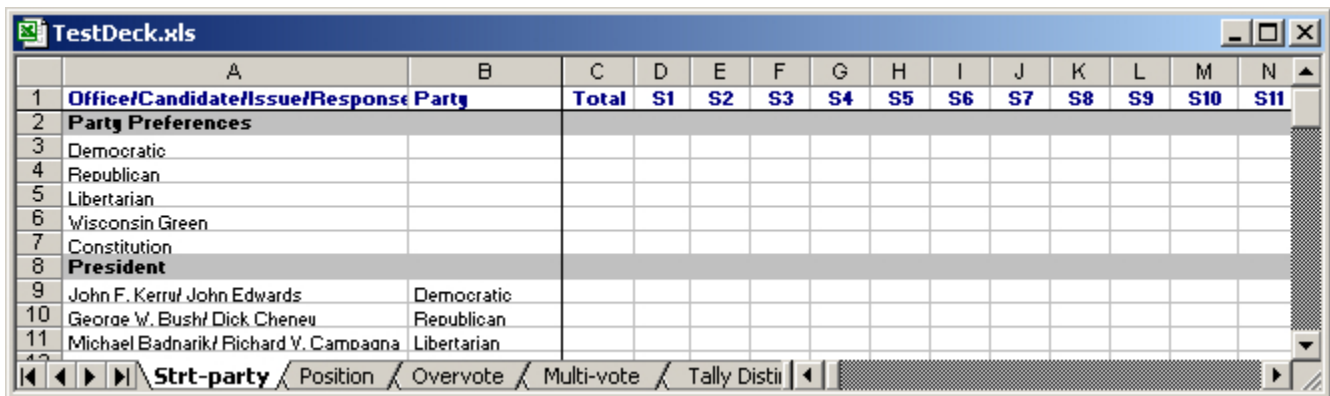
Sub-deck	Name	Description
1	Straight-party test	One ballot for each combination of one straight-party vote and one other mark.
2	Position test	Ballots with one vote for each possible candidate and each possible response to an issue.
3	Over-vote test	Ballots testing for over-votes in each contest.
4	Multi-vote test	Ballots for testing multi-vote contests.
5	Tally distinction ballots	Ballots to ensure that each candidate and issue receives a unique number of votes.

### Set up spreadsheets for the script of test ballots:

We used Excel. You could use another program, or generate the spreadsheets by hand on paper. The sample spreadsheets used in these guidelines can be found at locations listed in Appendix F.

For each sub-deck, create a separate spreadsheet, using the following format. Create the spreadsheet format once for the first sub-deck and then copy it exactly to the others. Use each spreadsheet to create the script for the associated sub-deck.

1. List the offices, candidates, issues, and responses in the first column, in the order they appear on the ballot, with party designations in the second column.
2. Label the third column "Total".
3. Number the remaining columns across the sheet to indicate the test ballot number, using prefixes to distinguish between ballots in the different sub-decks – such as S, P, V, M, and T.
4. Shade the rows that define offices to make sure you don't enter any votes for them in the script.



## Sub-Deck 1: Straight-Party Test Deck

Create this sub-deck **only** if your ballots include straight-party preference voting or Pick-A-Party ballots on which all candidates for partisan primaries are printed on the same physical ballot.

This is by far the largest sub-deck, and the most difficult to calculate the total votes for each candidate. Once you have completed this sub-deck, the others go much more quickly.

For each straight-party choice, you will create one test ballot with a single mark for that choice, AND a set of ballots with the straight-party choice marked and one other selection that is not a straight-party choice. For our sample ballot, there are five straight-party choices, and a total of 40 choices that are not straight-party choices (candidates, write-ins, or responses to issues). So, for the sample ballot this sub-deck would contain  $5 + (5 * 40) (= 205)$  test ballots.

### Define the scrip for the test ballots on the spreadsheet:

1. Mark each straight-party choice on a different ballot, and mark no other choices.
2. Follow these instructions for every straight-party choice.
  - a. Mark the straight-party choice.
  - b. Mark one selection that is not a straight-party choice.
  - c. Repeat steps a and b for a new test ballot until all selections that are **not** straight-party choice have been selected on a unique test ballot.

For example, the top left part of the spreadsheet would look like this:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Office/Candidate/Issue/Response Party		Total	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
2	<b>Party Preferences</b>													
3	Democratic			1					1	1	1	1	1	1
4	Republican				1									
5	Libertarian					1								
6	Wisconsin Green						1							
7	Constitution							1						
8	<b>President</b>													
9	John F. Kerry/ John Edwards	Democratic							1					
10	George W. Bush/ Dick Cheney	Republican								1				
11	Michael Badnarik/ Richard Y. Campagna	Libertarian									1			
12	David Cobb/ Patricia LaMarche	Wisconsin Green										1		
13	Ralph Nader/ Peter Miquel Camejo	Independent											1	
14	James Harris/ Margaret Trowe	Independent												1
15	Walter F. Brown/ Maru Alice Herbert	Independent												
16	Write-In													
17	<b>US Senate</b>													
18	Russ Feingold	Democratic												
19	Tim Michels	Republican												
20	Write-In													

### Calculate the number of valid votes for each ballot position:

**NOTE:** This calculation depends on your state laws regarding straight-party voting. For example, if a voter marks the “Democratic” straight-party choice and then votes for a Republican for State Senate, should the State Senate vote override the straight-party choice, should it not be counted, should it be counted as Democratic, or should the ballot be rejected?

- In the "Total" column for each candidate, enter the number equal to:
  - The number of straight-party choices for the candidate's party
  - Plus (+) the number of additional votes the candidate should receive from specific votes for the candidate when a different straight-party was voted. For Pick-A-Party primary ballots, this would be zero, as such cross-over voting is not permitted and such ballots should be rejected.
  - Minus (-) the number of straight-party votes the candidate should **not** receive because a vote for a different candidate overrode a straight-party vote for the candidate's party, or because such votes are counted as over-votes.
- In the "Total" column for each issue response, use the "SUM" formula to determine the total number of votes the response should receive from the test ballots in this sub-deck. If an issue has two possible responses, each of these should be equal to half the number of ballots in the sub-deck.

For example, on the sample ballot with five party choices and 40 other choices, there will be 41 test ballots with the "Democratic" party choice marked. So, in a state where a vote for a specific candidate overrides a party choice, in the "Total" column for Kerry/Edwards, enter:

41 straight-party votes for the Democratic party (1 + # of ballot lines)  
 + 4 Kerry/Edwards votes with other party choices marked (# of parties - 1)  
 - 7 (other candidates marked on the Democratic preference ballots (# of candidates -1))  
**38 Total**

An example for a candidate or write-in who gets no votes from straight-party votes:

0 straight-party votes for the candidate's party (always 0)  
 + 5 specific votes for this candidate with other party choices marked (# of parties)  
 - 0 other candidates marked on ballots with this candidate's party preference (always 0)  
**5 Total**

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Office/Candidate/Issue/Response Party		Total	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
2	<b>Party Preferences</b>													
3	Democratic			1					1	1	1	1	1	1
4	Republican				1									
5	Libertarian					1								
6	Wisconsin Green						1							
7	Constitution							1						
8	<b>President</b>													
9	John F. Kerry/ John Edwards	Democratic	38						1					
10	George W. Bush/ Dick Cheney	Republican	38							1				
11	Michael Badnarik/ Richard Y. Campagna	Libertarian	38								1			
12	David Cobb/ Patricia LaMarche	Wisconsin Green	38									1		
13	Ralph Nader/ Peter Miquel Camejo	Independent	5										1	
14	James Harris/ Margaret Trowe	Independent	5											1
15	Walter F. Brown/ Maru Alice Herbert	Independent	5											
16			5											

- Highlight any of the ballots that should be rejected, so you can count them easily. For example:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Office/Candidate/Issue/Response Party		Total	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
2	<b>Party Preferences</b>													
3	Democratic			1					1	1	1	1	1	1
4	Republican				1									

## Sub-Deck 2: Position Test Deck

Always create this sub-deck. The test ballots in this sub-deck verify that the machine (scanner, ballot-marker, or DRE) is detecting expected selections. They also verify that the scanner is *not* detecting marks anywhere else on the ballot or that the touch screen (ballot-marker or DRE) is *not* detecting touches anywhere else on the screen.

### Define the script for the test ballots on the spreadsheet:

1. On the first ballot, make no selections. This ensures that scanners are not reading pre-printed areas of the ballot or that default information is not registered on ballot-markers or DREs by mistake. This is the zero-mark pristine ballot.

For ballot-markers and DREs: To enter the “zero-mark pristine ballot,” page through each screen, making no selection for any candidate or question. Then, when the review screen is presented, attempt to cast the “unmarked” ballot.

2. For optical scan ballots: On the second ballot, blacken all areas of the ballot except the registration marks on the edges and the designated marking locations of the printed ballot. This ballot verifies that the software is not reading marks in undesigned areas. This is the “blackout ballot.”

For touch screens (ballot-markers and DREs): Ideally, the “blackout ballot” would be entered by touching every possible location not designated for voter touches, and making all those touches on every page presented by the DRE to a voter.

Clearly, this is impractical. So, for DRE systems, the *blackout ballot* consists of touching and double touching every screen in 8 places: all corners and the midpoints between each set of corners, as indicated by the red squares in this illustration.

Note: This ballot is not included in the test script for DREs that do not have a touch screen.



The remainder of the test ballots ensure that the scanner is reading marks in the proper locations or that the ballot-marker or DRE is correctly registering selections at the designated areas.

2. On the third ballot, select the first candidate or issue response in each contest.
3. On the fourth ballot, select the second choice in each contest.
4. On the fifth ballot, select the third choice in each contest. If a contest only has two choices, do not vote in that contest on this ballot.
5. Continue with this pattern until you have selected the last candidate in the contest with the most candidates.

The number of test ballots in this sub-deck is two more than the number of candidates in the contest with the most candidates. In our example, the sub-deck would contain 10 ballots — two 0-mark ballots and 8 ballot line marks on the Presidential race.

**Calculate the number of valid votes for each ballot position:**

In the “Total” column for each candidate and issue response, use the “SUM” formula to add the votes from all the test ballots.

- For Pick-A-Party ballots, marks should be handled according to your state’s laws. For example, in some states marks for partisan offices are not counted if the party choice is not marked.
- For all ballots EXCEPT Pick-A-Party ballots, all totals should be “1”.
- In all states, zero-mark ballots and black-out ballots should be rejected, so highlight them.

For example:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Office/Candidate/Issue/Response	Party	Total	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
2	<b>Party Preferences</b>												
3	Democratic												
4	Republican												
5	Libertarian												
6	Wisconsin Green												
7	<b>Constitution</b>												
8	<b>President</b>												
9	John F. Kerry/ John Edwards	Democratic	1			1							
10	George W. Bush/ Dick Cheney	Republican	1				1						
11	Michael Badnarik/ Richard V. Campagna	Libertarian	1					1					
12	David Cobb/ Patricia LaMarche	Wisconsin Green	1						1				
13	Ralph Nader/ Peter Miquel Camejo	Independent	1							1			
14	James Harris/ Margaret Trowe	Independent	1								1		
15	Walter F. Brown/ Maru Alice Herbert	Independent	1									1	
16	Write-In		1										1
17	<b>US Senate</b>												
18	Russ Feingold	Democratic	1			1							
19	Tim Michels	Republican	1				1						
20	Arif Khan	Libertarian	1					1					
21	Eugene A. Hem	Independent	1						1				
22	Write-In		1							1			
23	<b>US Congress, District No. 5</b>												
24	Bruan Kennedy	Democratic	1			1							
25	F. James Sensenbrenner, Jr.	Republican	1				1						
26	Tim Peterson	Libertarian	1					1					
27	Write-In		1							1			
28	<b>State Senate, District No. 9</b>												

### Sub-Deck 3: Over-vote Test Ballots

In this sub-deck, test ballots contain every combination of 2-mark over-votes, omitting all party preference selections.

#### Define the scrip for the test ballots on the spreadsheet:

For each contest, select every combination of two choices, with each pair of votes on a separate ballot. Do not include votes for party preferences. To accomplish this:

1. Mark the **first** candidate on **X-1** ballots, where X is the number of choices in the contest, and on those same ballots, mark a different ballot for each remaining candidate on the list.
2. Mark the **second** candidate on the next **X-2** ballots, and mark one of those same ballots for each remaining candidate.
3. Mark the **third** candidate on the next **X-3** ballots, and mark one of those same ballots for each remaining candidate.

The total number of ballots required for this sub-deck depends on the number of contests and the number of choices in each contest.

For example, a contest with 8 candidates requires  $(8 * 7)/2$ , or 28 ballots.

A second contest with 5 candidates requires an additional  $(5 * 4)/2$ , or 10 ballots.

A contest with 2 choices requires only  $(2*1)/2$ , or 1 ballot.

Our sample ballot contains 12 contests with a total of 40 choices among the contests as:

8, 5, 4, 3, 2, 2, 6, 2, 2, 2, 2, 2.

So this sub-deck would contain:  $28 + 10 + 6 + 3 + 1 + 1 + 15 + 1 + 1 + 1 + 1 + 1 = 69$  ballots

#### Calculate the number of valid votes for each ballot position:

In the "Total" column, enter "0" for every candidate and issue response for which only one vote is allowed. For any contests that allow the voter to vote for 2 candidates, enter the "SUM" of the votes for that candidate on all ballots. All these ballots should be rejected, **except those marked with two votes for a contest in which the voter is allowed to "Vote for 2" or more** – as in the seventh contest on the sample ballot discussed on page 14).

If there are many multi-vote contests, you could either highlight the ballots that should be rejected or use a new color to highlight those that should not be rejected.

The illustration below shows the first 18 ballots to create for the Presidential race on the sample ballot:

	A	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Office/Candidate/Issue/Response	Total	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12	Y13	Y14	Y15	Y16	Y17	Y18
8	<b>President</b>																			
9	John F. Kerry/ John Edwards	0	1	1	1	1	1	1	1											
10	George W. Bush/ Dick Cheney	0	1							1	1	1	1	1	1					
11	Michael Badnarik/ Richard V.	0		1						1						1	1	1	1	1
12	David Cobb/ Patricia LaMarche	0			1						1					1				
13	Ralph Nader/ Peter Miquel Camejo	0				1						1					1			
14	James Harris/ Margaret Trowe	0					1						1					1		
15	Walter F. Brown/ Maru Alice Herbert	0						1						1						1
16	Write-In	0								1					1					
17																				
18																				

### Sub-Deck 4: Multi-Vote Test Ballots

Create the script for this sub-deck **only** if your ballots include at least one contest in which a voter may cast **more than one valid vote**. This sub-deck tests that the software correctly counts each ballot with a valid number of votes on a multi-vote contest (such as a contest that instructs the voter to “Vote for 3”). It also tests that ballots containing more than the valid number are rejected.

#### Define the scrip for the test ballots on the spreadsheet:

For each multi-vote contest, create ballots with every possible combination of allowable votes and ballots with every possible combination of one over-vote in the contest. To do this:

1. Determine the maximum marks you will make on any ballot. This will always be one more the maximum number of allowed votes for the contest. For example, in a “Vote for 3” contest, you would make a maximum of four selections on any test ballot. In a “Vote for 5” contest, you would make a maximum of six selections.
2. Since all 1-mark and 2-mark ballots for this race have been created, begin with the 3-mark ballots. This is necessary for all “Vote for 2” and all “Vote for 3” contests. Mark each ballot with a different combination of 3 marks.

Below is an example of all possible ballots with three marks in a contest with six choices. Shading is provided in the illustration to help you identify the method we used to ensure that all combinations were selected.

	A	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	Office/Candidate/Issue/Respon	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20
38	Town Council																				
39	(vote for no more than 3)																				
39	James Adams	1	1	1	1	1	1	1	1	1	1										
40	Susan Baxter	1	1	1	1							1	1	1	1	1	1				
41	Anne Carter	1				1	1	1				1	1	1				1	1	1	
42	Write-In		1			1			1	1		1		1	1			1	1	1	1
43	Write-In			1			1			1	1		1		1			1	1	1	1
44	Write-In				1			1		1	1		1		1			1	1	1	1

Appendix A explains in detail our method for making all necessary selections for multi-mark ballots. In general, the method is to divide the ballots into groups, such that you can assign a single candidate votes on every ballot in the group, and then assign votes to the remaining candidates using the method of marking two (2) votes on each ballot.

In the first group (above) the first candidate receives votes on all ballots. In the second group, the first candidate gets no votes and the second candidate receives votes on all ballots. In the third group, neither the first nor the second candidate receives votes, and the third candidate receives votes from all ballots. And so on.

Note: The calculation for determining the number of ballots required in this situation is:

$$\frac{6 * 5 * 4}{3 * 2} = \frac{120}{6} = 20$$

The general formula is provided in Appendix A.



4. Generate the 4-mark ballots: Mark each ballot with a different combination of 4 marks.

Below is an example of all possible ballots with four marks in a contest with six choices. Shading is provided to help you distinguish the pattern used to ensure that all combinations were selected. Appendix A explains in detail our method for making all necessary selections for multi-mark ballots.

	A	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL
1	Office/Candidate/Issue/Respon	M21	M22	M23	M24	M25	M26	M27	M28	M29	M30	M31	M32	M33	M34	M35
38	<b>Town Council</b> <b>(vote for no more than 3)</b>															
39	James Adams	1	1	1	1	1	1	1	1	1						
40	Susan Baxter	1	1	1	1	1	1				1	1	1	1	1	
41	Anne Carter	1	1	1				1	1	1		1	1	1	1	1
42	Write-In	1			1	1		1	1		1	1	1		1	1
43	Write-In		1		1	1		1	1		1	1	1	1	1	1
44	Write-In			1	1	1		1	1		1	1	1	1	1	1

Note: The calculation for determining the number of ballots required in this situation is:

$$\frac{6 * 5 * 4 * 3}{4 * 3 * 2} = \frac{360}{24} = 15$$

The general formula is provided in Appendix A.

**Calculate the number of valid votes for each ballot position:**

In our sample contest that allows the voter to “Vote for 3”, all the 3-mark ballots should be counted as valid votes for each candidate, and all the 4-mark ballots should be rejected as over-votes.

To determine the number of valid votes for each candidate, use the “SUM” formula in the “Total” column to add all the **3-mark** ballots, and **not** the 4-mark ballots. For example:

	A	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Office/Candidate/Issue/Respon	Total	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18
38	<b>Town Council</b> <b>(vote for no more than 3)</b>																			
39	James Adams	10	1	1	1	1	1	1	1	1	1									
40	Susan Baxter	10	1	1	1	1						1	1	1	1	1	1	1		
41	Anne Carter	10	1				1	1	1			1	1	1					1	1
42	Write-In	10		1			1			1	1		1		1	1		1	1	1
43	Write-In	10			1			1			1		1		1		1		1	1
44	Write-In	10				1			1		1		1		1		1		1	1

Highlight the entire column for all ballots with more than the allowable number of votes, since they will be rejected.

## Sub-Deck 5: Tally Distinction Test Ballots

In every complete test deck, each candidate and issue response must receive a unique number of votes to ensure that votes from one are not being switched to another. This sub-deck adds the necessary ballots to ensure the uniqueness of the votes each ballot position receives.

### Define the script for the test ballots on the spreadsheet:

1. Rename the “Total” column on the spreadsheet for this test deck to “SD 1-4” and use it to calculate the total votes for each ballot position in Sub-Decks 1 through 4. In Excel the formula to use in cell C9, for example, will reference the same cell in each of the other spreadsheet. For our sample spreadsheet, the formula would be:

**='Strt-party'!C9+Position!C9+Over-vote!C9+'Multi-vote'!C9**

The number of votes for each position will depend on:

- Whether your ballots allow straight-party votes, how over-votes in them are counted, and
- Whether an office allows more than one vote per office.

The following table shows the total votes in our sample ballot – before adding Tally Distinction ballots:

Office or Issue	Selection	Votes in Sub-Deck		
		SD1-4	Add	Total
President	John F. Kerry/ John Edwards	39		
	George W. Bush/ Dick Cheney	39		
	Michael Badnarik/ Richard V. Campa	39		
	David Cobb/ Patricia LaMarche	39		
	Ralph Nader/ Peter Miguel Camejo	6		
	James Harris/ Margaret Trowe	6		
	Walter F. Brown/ Mary Alice Herbert	6		
	Write-In	6		
US Senate	Russ Feingold	42		
	Tim Michels	42		
	Arif Khan	42		
	Eugene A. Hem	6		
	Write-In	6		
US Congress, District No. 5	Bryan Kennedy	43		
	F. James Sensenbrenner, Jr.	43		
	Tim Peterson	6		
	Write-In	6		
State Senate, District No. 8	Jennifer Morales	44		
	Alberta Darling	44		
	Write-In	6		
State Assembly, District No. 24	Sue Jeskewitz	45		
	Write-In	6		
Proposition A	Yes	6		

Office or Issue	Selection	Votes in Sub-Deck		
		SD1-4	Add	Total
	No	6		
Town Council	James Adams	57		
(vote for no more than 3)	Susan Baxter	57		
	Anne Carter	57		
	Write-In	21		
	Write-In	21		
	Write-In	21		
Proposition B	Yes	6		
	No	6		
Proposition C	Yes	6		
	No	6		
Proposition D	Yes	6		
	No	6		
Proposition E	Yes	6		
	No	6		
Proposition F	Yes	6		
	No	6		

2. Insert two more columns after the "SD 1-4" column.
3. Label the first one "Add."
4. Label the second inserted column "Total" and for each ballot line, add the votes from the "SD1-4" column and the "Add" column.
5. In the "Add" column for each ballot position, enter the number of votes required in Sub-Deck 5 to ensure that each position receives a unique number of votes.

The number of ballots in this sub-deck will equal the largest total number of additional votes to be added in any contest. For our sample ballot, this would be Proposition F, so the sub-deck would contain 41 ballots. Note that this number is not the smallest possible; you can work with the "Add" column to minimize the number of ballots, if you need to.

Office or Issue	Selection	Votes in Sub-Deck		
		SD1-4	Add	Total
President	John F. Kerry/ John Edwards	39	0	39
	George W. Bush/ Dick Cheney	39	1	40
	Michael Badnarik/ Richard V. Campa	39	2	41
	David Cobb/ Patricia LaMarche	39	3	42
	Ralph Nader/ Peter Miguel Camejo	6	0	6
	James Harris/ Margaret Trowe	6	1	7
	Walter F. Brown/ Mary Alice Herbert	6	2	8
	Write-In	6	3	9
US Senate	Russ Feingold	42	1	43
	Tim Michels	42	2	44

Office or Issue	Selection	Votes in Sub-Deck		
		SD1-4	Add	Total
	Arif Khan	42	3	45
	Eugene A. Hem	6	4	10
	Write-In	6	5	11
US Congress, District No. 5	Bryan Kennedy	43	3	46
	F. James Sensenbrenner, Jr.	43	4	47
	Tim Peterson	6	6	12
	Write-In	6	7	13
State Senate, District No. 8	Jennifer Morales	44	4	48
	Alberta Darling	44	5	49
	Write-In	6	8	14
State Assembly, District No. 24	Sue Jeskewitz	45	5	50
	Write-In	6	9	15
Proposition A	Yes	6	10	16
	No	6	11	17
Town Council	James Adams	57	0	57
(vote for no more than 3)	Susan Baxter	57	1	58
	Anne Carter	57	2	59
	Write-In	21	7	28
	Write-In	21	8	29
	Write-In	21	9	30
Proposition B	Yes	6	12	18
	No	6	13	19
Proposition C	Yes	6	14	20
	No	6	15	21
Proposition D	Yes	6	16	22
	No	6	17	23
Proposition E	Yes	6	18	24
	No	6	19	25
Proposition F	Yes	6	20	26
	No	6	21	27

6. Insert a new column after the "Total" column, and label it "SD5". In each cell, enter the formula to sum the values in the cells in all the columns to the right. Then as you add votes in the cells, you can see how many have been added.
7. Enter marks on the ballots as necessary to give each ballot position the quantity in the "Add" column, that is, until the "SD5" column matches the "Add" column. As you add ballots, make sure that there are no over-votes on any of the ballots in sub-deck 5.

# Using the Test Ballots

## ***Executing the Test***

1. Assign each ballot in the script a unique ballot number.
2. Set up the form in Appendix D and complete all information about the expected results.
- 3a. If you are testing an optical scanner, mark a separate paper ballot for each test ballot defined in the scripts. Have a co-worker proof the ballots and compare them to the script to ensure that the deck is created according to the script. Then run the ballots through the scanner.
- 4b. If you are testing a DRE or an electronic ballot marking device, such as the AutoMark, enter each ballot by hand, in ballot-number order, making selections on the screen as a voter would. Have a co-worker watch as you enter the ballots, to ensure that they are entered according to the script.

Note: It is very important to test for likely AND unlikely behaviors of voters. Use the following subsections to make sure you test for unlikely behaviors.

## **For Optically Scanned Ballots**

1. Voters might deposit their ballots in any orientation in a precinct scanner. Therefore, present test ballots in all four possible orientations: Face-up/Top First, Face-up/Bottom First, Face-down/Top First, and Face-down/Bottom First.
2. Voters, especially absentee voters, might use any marking instrument to complete their ballot. If any of these ballots fail, you may need to recalibrate the scanner to detect them. To test for this possibility:
  - Create some of the test ballots using a marking device other than the one provided in the polling place, for example, different colored pens or #2 and #3 pencils.
  - For fill-in-the-oval type ballots, create a few ballots marked differently than a well-filled oval, for example check marks, X's, lightly filled ovals, filled ovals with a void in the center, filled beyond the bounds of the oval.
  - For connect-the-arrow-head-to-the-tail type, create a few ballots with wavy lines. Some electors may have a tremor which makes a single straight line difficult.
  - For polling places hosting multiple precincts or multiple wards, there is no guarantee the elector will get the ballot for his ward, or that the elector will deposit the ballot in the correct scanning unit. Try to feed a ballot for Ward 2 through the voting machine programmed for Ward 1. The expected behavior is to reject ballots from other wards.

## **For DREs or Other Electronic Devices**

1. As you enter test-script ballots onto the screen, try all the actions a voter might take – either accidentally or on purpose. For example:
  - Use the “Previous” function to go back to the previous contest and change the vote.
  - Skip a contest that should have a vote, move ahead, and then go back to the contest and vote it.
  - Try the “Help” function from several different screens.
  - Use the write-in feature to write in several candidates. Edit the name once it has been entered.

2. For over-voted ballots, such as those in the Over-vote Sub-Deck and the over-voted ballots in the Multi-Vote Sub-Deck, touch each position to make sure you are warned of your attempt to over-vote.
3. Check the review screen **very carefully** against the script for each ballot. An entry error can cause you to obtain unexpected results, which may be difficult to diagnose. AND, once the problem is found, you may need to execute the test again, which can be very time-consuming.
4. Consider videotaping the process of entering ballots on the screens. If the results are not what you expect, you can check the video to determine whether there was an entry error or the machine recorded or tallied incorrectly.

## **For Touch-Screens**

As you enter ballots on a touch screen, test for as many possible unlikely or unusual voter behaviors as you can think of. For example:

- Touch the screen in three different ways; lightly, hard and with a tremor
- Touch the screen at the edge of a “button”, such as a line of the rectangle indicating a button.
- Touch outside the “buttons” on the screen. For example, press slightly above, below, and to each side of each button.
- Touch the screen with very dry (use isopropyl alcohol) fingers and damp fingers in order to simulate the variations in electrical properties of skin.
- Touch the screen with some part of your hand other than your finger tip, such as the back of a knuckle or a long finger nail. Touch it with something other than your finger, for example a pencil. Some voters may use a mouth stick or head stick.
- If a stylus is provided, touch the screen with a finger, and with something other than the stylus, for example, a pen or a pencil.
- Touch two voting area simultaneously
- Double-touch or double-click on the button. Many user interfaces differentiate between single and double clicks. Single and double clicks should work the same.

## **For Units Intended for Use by People with Disabilities**

Enter at least one ballot from each sub-deck using EACH feature intended for people with disabilities. This means entering one ballot from each sub-deck using the audio feature, one from each sub-deck using sip-and-puff, one from each sub-deck using a joystick, and any other feature provided on the machine. Make sure to enter write-in names in each case.

Ideally, have people with disabilities enter the ballots. If this is not feasible, make sure you cover the screen with a barrier, such as a piece of cardboard, when you are entering audio ballots.

## **For Foreign Language Units**

Enter at least one ballot from each sub-deck using EACH foreign language provided on the unit. Ideally, have people who speak the language enter the ballots.

## ***Recording Test Results***

Before executing a test, use the form in Appendix E to set up the form on which you will create a complete record of the testing and the results. Include the following:

1. The date the test was executed.
2. The name or description of the test performed.
3. The name of the person or persons who performed the test and recorded the results.
4. The version number of the software under test.
5. The model number of the machine on which the test was executed.
6. The serial number of the machine on which the test was executed.
7. The test result – either “pass” if the results match the expected results exactly, or “fail” if there is even one discrepancy.
8. If the software failed, record as much information about the failure as you can, including the ballot position with an incorrect number of votes, the number of the ballot which was incorrectly counted (if you know it), the number of the ballot that should have been rejected and was not or that was rejected and should not have been.
9. The signature of the person or persons who performed the test and recorded the results.

## ***Checking the Accuracy of the Results***

After executing the test:

1. Complete the information in the table on the Results Form (Appendix E)
2. Compare the computer-generated results to the results you predicted on the Expected Results Form (Appendix D), and determine whether the machine passed or failed. ANY discrepancies indicate a failure and must be investigated. Once the problem is discovered and repaired, a re-test must be performed.

Notes:

- If an attempt to over-vote on a DRE did not generate an error message or some other type of warning to the voter, this constitutes a failure.
- If the review screen on a DRE does not reflect the choices entered on the screens, this constitutes a failure.



## **Determining the Cause of Failures**

The goal for your test plan is to find defects due to mistakes. But, even with this narrow focus, in addition to discovering ballot-programming defects, you might even uncover defects in the underlying software. What if the software fails the test? It is crucial to follow the evidence where ever it may lead. These tips may help you determine the cause and how to repair it.

- **The ballot definition may not take into account the ballot rotation.**

This occurred in Pottawattamie County, Iowa in the June 2006 primary election. When the error was corrected, outcomes of nine elections changed.

- **Straight-party votes may not be handled correctly.**

In November 2002, ballot scanners in Wayne County, North Carolina eliminated all straight-party votes from the results. Adding them changed the outcome for a state representative contest.

In November 2004, ballot scanners in Taylor County, Wisconsin failed to count all straight-party votes – over 25% of the ballots. The error was discovered 4½ months after the election, too late to correct the results.

In November 2005, a programming error in Cumberland County, Pennsylvania gave straight-party Democratic votes to Republican, and didn't count straight-party Republican votes at all.

- **Party designations may not be correct in the ballot definition.**

In November 2000, ballot scanners in Bernalillo County, New Mexico failed to count votes because the programmer had not linked the candidates' names to their respective parties.

- **Errors may occur when results from scanners and DREs are combined.**

In April 2002, in Miami-Dade County, Florida, software used to combine absentee votes with electronic ballots changed the order of the candidates' names as it computed the results. The initial tally showed wins for two City Council candidates who actually lost the election.

- **A candidate may be omitted from the ballot definition for the proper jurisdiction.**

In November 2004, DREs in Maryland failed to display some candidate names on the screen.

**Note:** These are only some of the most common errors. They are intended only to help you troubleshoot, but they are not an exhaustive list of the potential causes of discrepancies between your electronic results and the test ballots.

For examples of specific errors that could have been detected by thorough pre-election testing, see:

<http://www.votersunite.org/info/Vote-Switchinginthenews-byState.pdf>

For additional examples of voting system errors that have been reported in the news, see:

- *Malfunctions and Miscounts, Sorted by Vendor*

<http://www.votersunite.org/info/messupsbyvendor.asp>

- *Malfunctions and Miscounts, Sorted by State*

<http://www.votersunite.org/info/previousmessups.asp>

## Appendix A. Creating Scripts for Multi-Mark Test Ballots

This appendix contains instructions for constructing a script for multi-mark ballots when you are including all possible combinations of the two marks. This method explained in this appendix applies only to the Over-vote Sub-Deck and the Multi-Vote Sub-Deck.

Note that the method is not necessary for the Straight-Party sub-deck, even though most of the ballots contain two marks. On the Straight-Party sub-deck, each of the two-mark ballots has a vote for a party preference and one other mark; it is not necessary to generate all possible combinations of two marks.

### Constructing a Script for Over-vote Test Ballots

Use these instructions to create the script for the Over-vote Sub-Deck. Creating this script is fairly easy, since each ballot contains only two marks and the combinations are simple.

Create a separate set of test ballots for each contest on the ballot. Begin with the first contest on the ballot, and move through the remaining contests in order. Follow the instructions below, making sure that you **place the marks for each contest in different columns**, so they will be marked on different ballots.

**Tip:** If you are using Microsoft Excel, use the “Freeze Panes” feature to ensure that the list of contests and candidates remains in view even when you are entering marks in the columns far to the right of the candidate list. To do this, select column “B”, then select the **Window** menu, **Freeze Panes** command. Or, if you want to keep the column headings in view as well, select cell “B2” then select the Freeze Panes command.

1. Determine the list of the ballot positions to be included in the set.  
For example:

If you are creating the set of over-voted ballots for the Presidential contest in our sample ballot, the list would include all the candidates and the write-in.

If you are creating the set of ballots for a proposition, the list would include the optional responses for that proposition.

President
John F. Kerry/ John Edwards
George W. Bush/ Dick Cheney
Michael Badnarik/ Richard Y. Campagna
David Cobb/ Patricia LaMarche
Ralph Nader/ Peter Miquel Camejo
James Harris/ Margaret Trowe
Walter F. Brown/ Maru Alice Herbert
Write-In

Proposition A
Yes
No

2. Mark the first candidate on each ballot up to the number of candidates minus 1. On our sample, this would cast 7 ballots for Kerry/Edwards.
3. On those same ballots, mark each other candidate on a separate ballot, starting with the second candidate and moving down the list in order.

President							
John F. Kerry/ John Edwards	1	1	1	1	1	1	1
George W. Bush/ Dick Cheney	1						
Michael Badnarik/ Richard Y. Campagna		1					
David Cobb/ Patricia LaMarche			1				
Ralph Nader/ Peter Miquel Camejo				1			
James Harris/ Margaret Trowe					1		
Walter F. Brown/ Maru Alice Herbert						1	
Write-In							1

4. At this point, all the ballots that should include a mark for Kerry/Edwards have been marked. Repeat steps 2 and 3, **AS IF** there were only seven candidates beginning with Bush/Cheney. Note that this subset will contain six ballots (the number of candidates minus 1).

President						
John F. Kerry/ John Edwards						
George W. Bush/ Dick Cheney	1	1	1	1	1	1
Michael Badnarik/ Richard Y Campagna	1					
David Cobb/ Patricia LaMarche		1				
Ralph Nader/ Peter Miquel Cameio			1			
James Harris/ Margaret Trowe				1		
Walter F. Brown/ Maru Alice Herbert					1	
Write-In						1

5. Repeat steps 2 and 3 again, **AS IF** there were only six candidates beginning with Badnarik/Campagna.

President					
John F. Kerry/ John Edwards					
George W. Bush/ Dick Cheney					
Michael Badnarik/ Richard Y Campagna	1	1	1	1	1
David Cobb/ Patricia LaMarche	1				
Ralph Nader/ Peter Miquel Cameio		1			
James Harris/ Margaret Trowe			1		
Walter F. Brown/ Maru Alice Herbert				1	
Write-In					1

6. Repeat this method starting with the fourth, fifth, sixth, and seventh candidate on the list. Each subset contains one fewer ballots.

President							
John F. Kerry/ John Edwards							
George W. Bush/ Dick Cheney							
Michael Badnarik/ Richard Y Campagna	1	1	1	1			
David Cobb/ Patricia LaMarche	1						
Ralph Nader/ Peter Miquel Cameio		1			1	1	1
James Harris/ Margaret Trowe			1			1	1
Walter F. Brown/ Maru Alice Herbert			1		1	1	1
Write-In				1		1	1

7. To create the complete Over-vote Sub-Deck, repeat this same procedure for the choices in each contest on the ballot, moving down through the contests in order.

### Constructing a Script for Multi-vote Test Ballots

Use these instructions to create the script for a contest in which the voter can vote for more than two candidates. Since each contest will contain 3, 4, or more marks, setting up all possible combinations is a bit more complex, but the method is logical, and once you understand the pattern, you'll see that it can be used in every case.

As you did for the Over-vote Sub-Deck, create a separate set of test ballots for each multi-vote contest on the ballot. Begin with the first contest on the ballot, and move through the other contests in order. Follow the instructions below, making sure that you **place the marks for each contest in different columns**, so they will be marked on different ballots.

**Tip:** If you are using Microsoft Excel, use the "Freeze Panes" feature to ensure that the list of contests and candidates remains in view even when you are entering marks in the columns far to the right of the candidate list. To do this, select column "B", then select the **Window** menu, **Freeze Panes** command. Or, if you want to keep the column headings in view as well, select cell "B2" then select the Freeze Panes command.

## Determining the Number of Ballots Needed for a Set of Multi-Mark Ballots

To determine the quantity needed for a single contest in which the voter can vote three or more times, use the following formula or the table below:

$$\frac{C!}{M!(C-M)!} \text{ where } C = \text{the number of choices in the contest, and } M = \text{the number of marks on each ballot}$$

The exclamation mark (!) means “factorial.” This mathematical function means to multiply the number before the “!” times each of the numbers smaller than it.

For example,  $6! = 6 * 5 * 4 * 3 * 2 * 1 = 720$

So in a contest with 6 candidates (including write-ins) where the voter is allowed to vote for 3 candidates you will need 3-mark and 4-mark ballots. The number of test ballots with exactly 3 marks among 6 possible ballot lines would be:

$$\frac{6!}{3!(6-3)!} = \frac{6*5*4*3*2*1}{3*2*1(3*2*1)} = \frac{720}{36} = 20$$

The number of test ballots with exactly 4 marks among 6 possible ballot lines would be:

$$\frac{6!}{4!(6-2)!} = \frac{6*5*4*3*2*1}{4*3*2*1(2*1)} = \frac{720}{48} = 15$$

For a total of 35 multi-vote ballots needed to test the vote for 3 from among 6 race found on the sample ballot.

The following table gives the number of ballots required in most situations you are likely to encounter.

Number of Marks to Make	Number of Candidates in the Contest (including write-in and None of the Above)									
	3	4	5	6	7	8	9	10	11	12
3	1	4	10	20	35	56	84	120	165	220
4		1	5	15	35	70	126	210	330	495
5			1	6	21	56	126	252	462	792
6				1	7	28	84	210	462	924
7					1	8	36	120	330	792
8						1	9	45	165	495
9							1	10	55	220

Generating any script is simplified by separating the script into groups defined as follows:

- The first group contains all the ballots with a vote for the first candidate.
- The second contains ballots without votes for the first candidate, but with votes for the second.
- The third contains ballots without votes for either the first or second, but with votes for the third.
- And so on.

The method for generating 3-mark ballots is used in all other multi-mark ballot generation. So, it is explained first.

## Generating a Script with 3-Mark Ballots

### Generating the First Group of 3-Mark Ballots

This group contains all the ballots with a vote for the first candidate in the contest. The process is this:

1. On each ballot in the first group, mark the first candidate.
2. On those same ballots, mark the remaining candidates **AS IF** you were generating 2-mark ballots for those candidates.

You can calculate the number of ballots in this group by using the following formula:

$$\frac{(C-1)*(C-2)}{2} \text{ where } C = \text{the number of choices.}$$

For our sample multi-vote contest, this is:  $5*4/2 = 20/2 = 10$

The ballot script looks like this:

Office/Candidate/Issue/Respon	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
James Adams	1	1	1	1	1	1	1	1	1	1
Susan Baxter	1	1	1	1						
Anne Carter	1				1	1	1			
Write-In		1			1			1	1	
Write-In			1			1		1		1
Write-In				1			1		1	1

### Generating the Second Group of 3-Mark Ballots

The second group contains ballots without any votes for the first candidate, but with a vote for the second. The method of creating it is this:

1. Give no marks to the first candidate.
2. Use the process for generating the FIRST group to create a group of 3-mark ballots for the remaining candidates in the list.

So, for our sample contest, it would be **AS IF** the first candidate were not there at all and you were generating 3-mark ballots for 5 candidates.

- a. On each ballot, mark the second candidate.
- b. On those same ballots, mark the remaining candidates **AS IF** you were generating 2-mark ballots for those candidates.

The number of ballots in this group would be:  $(4 * 3)/2 = 12/2 = 6$ .

Office/Candidate/Issue/Respon	M11	M12	M13	M14	M15	M16
James Adams						
Susan Baxter	1	1	1	1	1	1
Anne Carter	1	1	1			
Write-In	1			1	1	
Write-In		1		1		1
Write-In			1		1	1

### Generating the Third Group of 3-Mark Ballots

The second group contains ballots without any votes for the first or second candidates, but with a vote for the third. The method of creating it is this:

1. Give no marks to the first candidate or second candidate.
2. Again, use the process for generating the FIRST group to create a group of 3-mark ballots for the remaining candidates in the list.

So, for our sample contest, it would be **AS IF** the first candidate and second candidates were not there at all and you were generating 3-mark ballots for 4 candidates.

- a. On each ballot, mark the third candidate.
- b. On those same ballots, mark the remaining candidates **AS IF** you were generating 2-mark ballots for those candidates.

The number of ballots in this group would be:  $(3*2)/2 = 6/2 = 3$ .

Office/Candidate/Issue/Respon	M17	M18	M19
James Adams			
Susan Baxter			
Anne Carter	1	1	1
Write-In	1	1	
Write-In	1		1
Write-In		1	1

### Generating the Rest of the Groups of 3-Mark Ballots

Repeat this same process, until all candidates except the last three (3) receive no marks. In the case of our sample contest, this is the fourth group. This final group will have one ballot with marks for the last three candidates.

Office/Candidate/Issue/Respon	M20
James Adams	
Susan Baxter	
Anne Carter	
Write-In	1
Write-In	1
Write-In	1

The complete set of 3-mark ballots in a contest with 6 candidates looks like this:

Office/Candidate/Issue/Respon	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20
James Adams	1	1	1	1	1	1	1	1	1	1										
Susan Baxter	1	1	1	1							1	1	1	1	1	1				
Anne Carter	1				1	1	1				1	1	1				1	1	1	
Write-In		1			1			1	1		1			1	1		1	1		1
Write-In			1			1		1		1		1		1		1	1		1	1
Write-In				1			1		1	1			1		1	1		1	1	1

## Generating a Script with 4-Mark Ballots

The general concept for generating a set of 4-mark ballots is also based on groups of ballots. In each case you will use the instructions for generating 3-mark ballots.

### Group 1:

1. Mark the first candidate on each ballot.
2. On those same ballots, mark the remaining candidates **AS IF** you were generating **3-mark** ballots.

### Group 2:

1. Give no marks to the first candidate.
2. Mark the second candidate on each ballot.
3. On those same ballots, mark the remaining candidates **AS IF** you were generating **3-mark** ballots.

### Group 3:

1. Give no marks to the first or second candidate.
2. Mark the third candidate on each ballot.
3. On those same ballots, mark the remaining candidates **AS IF** you were generating **3-mark** ballots.

### Remaining groups:

Continue this process until all candidates except the last four (4) receive no marks. This final group will have one ballot with marks for the last four candidates.

For example:

Office/Candidate/Issue/Respon	M21	M22	M23	M24	M25	M26	M27	M28	M29	M30	M31	M32	M33	M34	M35
James Adams	1	1	1	1	1	1	1	1	1	1					
Susan Baxter	1	1	1	1	1	1					1	1	1	1	
Anne Carter	1	1	1				1	1	1		1	1	1		1
Write-In	1			1	1		1	1		1	1	1		1	1
Write-In		1		1	1	1	1	1	1	1		1	1	1	1
Write-In			1		1	1	1	1	1	1	1	1	1	1	1

## Generating a Script with M-Mark Ballots

This is the general process, where “M” is the number of marks to make on each ballot.

### Group 1:

1. Mark the first candidate on each ballot.
2. On those same ballots, mark the remaining candidates **AS IF** you were generating ballots with one mark less than M marks (**M-1**).

### Group 2:

1. Give no marks to the first candidate.
2. Mark the second candidate on each ballot.
3. On those same ballots, mark the remaining candidates **AS IF** you were generating ballots with one mark fewer than M (**M-1**).

### Remaining groups:

Continue this process until all candidates except the last M candidates receive no marks. This final group will have one ballot with marks for the last M candidates.

## Appendix B. Reporting Defects Found in these Guidelines

The sub-decks of test ballots described in the document are for ballot and election combinations which the author has encountered and many which the author has not encountered. To the author's knowledge, voting machinery in every jurisdiction in United States of America can be tested for programming mistakes with the 5 sub-decks contained in this document.

Testing for security or malicious code is beyond the scope of this document and such defects may not be detected by the 5 sub-decks contained in this document.

If you encounter an election which has a race on a ballot which these guidelines fail to test properly, please contact the author at [CutWaste@WashburnResearch.org](mailto:CutWaste@WashburnResearch.org). I want these guidelines to be correct and complete with a description of the race and ballot.

If you can think of a new sub-deck to create, the author would like to include this sub-deck into future revisions of this document. Please contact the author(s) at: [CutWaste@WashburnResearch.org](mailto:CutWaste@WashburnResearch.org).



## Appendix C: Sample Ballot

Adapted from District 1, Germantown, WI Ballot  
Of November 2, 2004

Party Preferences
Democratic
Republican
Libertarian
Wisconsin Green
Constitution

President	
John F. Kerry/ John Edwards	Democratic
George W. Bush/ Dick Cheney	Republican
Michael Badnarik/ Richard V. Campagna	Libertarian
David Cobb/ Patricia LaMarche	Wisconsin Green
Ralph Nader/ Peter Miguel Camejo	Independent
James Harris/ Margaret Trowe	Independent
Walter F. Brown/ Mary Alice Herbert	Independent
Write-In	

US Senate	
Russ Feingold	Democratic
Tim Michels	Republican
Arif Khan	Libertarian
Eugene A. Hem	Independent
Write-In	

US Congress, District No. 5	
Bryan Kennedy	Democratic
F. James Sensenbrenner, Jr.	Republican
Tim Peterson	Libertarian
Write-In	

State Senate, District No. 8	
Jennifer Morales	Democratic
Alberta Darling	Republican
Write-In	

State Assembly, District No. 24	
Sue Jeskewitz	Republican
Write-In	

Proposition A	
Yes	
No	

Town Council (vote for no more than 3)	
James Adams	Democratic
Susan Baxter	Republican
Anne Carter	Libertarian
Write-in	
Write-in	
Write-in	

Proposition B	
Yes	
No	

Proposition C	
Yes	
No	

Proposition D	
Yes	
No	

Proposition E	
Yes	
No	

Proposition F	
Yes	
No	

Copyright Notice.

This document retains full copyright protection to all of its authors jointly with the following exception. This document may be copied in any form for any purpose, provided the document is copied in its entirety, this notice of copyright is preserved, and all notices of authorship are preserved.

## Appendix D: Sample Form — Expected Results

Election: \_\_\_\_\_

Election Date: \_\_\_\_\_

Chief Creator of Test Deck: \_\_\_\_\_

Signature: \_\_\_\_\_

Test Deck	# of Test Ballots Expected to be ...			Comments / Notes
	Accepted	Rejected	Segregated	
Straight Party				
Positional				
Overvote				
Multi-vote				
Tally Distinction				
<b>Total</b>				

Other Creator of Test Deck: \_\_\_\_\_

Signature: \_\_\_\_\_

Other Creator of Test Deck: \_\_\_\_\_

Signature: \_\_\_\_\_

Other Creator of Test Deck: \_\_\_\_\_

Signature: \_\_\_\_\_

## Appendix E: Sample Form — Test Result

Create one copy of this form for each machine tested.

Election: \_\_\_\_\_

Precinct/Polling Place: \_\_\_\_\_

Election Date: \_\_\_\_\_

Testing Start Date: \_\_\_\_\_

Name of Chief Tester: \_\_\_\_\_

Signature of Chief Tester: \_\_\_\_\_

Vendor Name: \_\_\_\_\_

Machine Model Number: \_\_\_\_\_

Machine Serial Number: \_\_\_\_\_

Software Version: \_\_\_\_\_

Test Videotaped?  Yes  No If Yes, Tape Number \_\_\_\_\_

Test Deck	# of Test Ballots ...			Comments / Notes
	Accepted	Rejected	Segregated	
Straight Party				
Positional				
Overvote				
Multi-vote				
Tally Distinction				
<b>Total</b>				

Results compared with expected results?  Yes  No

Results Matched?  Yes  No If No, Date for Re-Test \_\_\_\_\_

## Appendix F: Where To Find This And Other Documents

A copy of these testing guidelines can be found at:

<http://www.washburnresearch.org/archive/TestingGuidelines/GuidelinesForCreatingTestBallots.pdf>

A single page, executive summary of these testing guidelines can be found at:

<http://www.washburnresearch.org/archive/TestingGuidelines/ExecutiveSummaryForTestGuidelines.pdf>

An archive of the Microsoft Excel spread sheets used in these testing guidelines can be found at:

<http://www.washburnresearch.org/archive/TestingGuidelines/SampleSpreadsheetsForTestGuidelines.zip>

An archive of the executive summary, these testing guidelines, and the sample spread sheets can be found at:

<http://www.washburnresearch.org/archive/TestingGuidelines/TestGuidelines-Everything.zip>