Voluntary Voting System Guidelines
Version 2

DRAFT

April 13, 2005

Table of Contents

VVSG Version 2 - DRAFT (April 2005)

1 Overview

Created in response to the Help America Vote Act (HAVA) of 2002 and based on the initial set of recommendations of the Technical Guidelines Development Committee (TGDC) mandated by HAVA, the Voluntary Voting System Guidelines -Version 2 (VVSG2) builds on the initial recommendations in the Voluntary Voting System Guidelines-Version 1(VVSG1). Likewise, VVSG2 builds on relevant voting standards and standards development efforts including the Voluntary Voting Standards (2002), the IEEE P1583 draft standards and the NIST Human Factors Report (NIST SP 500-256). *(Note: The VVSG Version 1 augments the Voting System Standard (VSS) of 2002, which was promulgated by the Federal Election Commission (FEC).*

In addition, VVSG2 will introduce a new organizational structure for voluntary voting standards, a voting process model, a standards maintenance program and a description of the connection between standards development and a testing program.

The three working sub committees of the Technical Guidelines Development Committee have provided input, reviewed and approved the organizational format for the new VVSG- Version 2 draft standards. The document in its current form (VVSG2 [April 2005]) is partially complete. Eight organizational principles describe a high-level framework for defining the functions of a voting system. The functions map into a voting process model. In this initial version of the draft standard, NIST has generated standards development work on a subset of the principles. A standards maintenance program and an approved voting systems testing program are not included in VVSG2 (April 2005). Later versions of this draft standard will also include a conformance clause.

The complete VVSG version 2 will be organized and written in a format geared to three main audiences: voting system and product developers; election officials; and testing laboratories. However, policy makers and interested voters should also find the document a useful reference.

1.1 Background

The Help America Vote Act (HAVA) established the Technical Guidelines Development Committee to assist the Election Assistance Commission with the development of voluntary voting system guidelines. HAVA directs the National Institute of Standards and Technology (NIST) to chair the TGDC and to provide technical support to the TGDC in the development of these guidelines. The TGDC's initial set of recommendations for these guidelines were due to the Election Assistance Commission in May 2005, in accordance with HAVA's nine-month deadline.

VVSG1 assists the states in preparing for the 2006 election. The document augments the 2002 VSS to address the critical areas of accessibility, usability and computer security. In addition, the VVSG1 includes an improved glossary to promote common understanding, a conformance clause,

and an updated Appendix on error rates. However, VVSG Version 1 is only an interim set of guidelines.

VVSG2 represents a long term continuous effort to create and maintain a redesigned VVSG that will address a large range of issues including rewriting requirements, where necessary, to make them more precise and testable, and further addressing key human factors and computer security issues. These issues affect the basic requirements of voting systems to such a degree that these types of changes cannot reasonably be made and tested in time for the 2006 election cycle.

1.2 Brief History of Voting Systems Standards and Guidelines

In 1975, the National Bureau of Standards (now the National Institute of Standards and Technology) and the Office of the Federal Elections (the Office of Election Administration's predecessor at the General Accounting Office) produced a joint report, *Effective Use of Computing Technology in Vote Tallying.* This report concluded that a basic cause of computer-related election problems was the lack of appropriate technical skills at the state and local levels to develop or implement sophisticated Standards against which voting system hardware and software could be tested. A subsequent Congressionally authorized study produced by the FEC and the National Bureau of Standards detailed the need for a federal agency to develop national performance Standards that could be used as a tool by state and local election officials in the testing, certification, and procurement of computer-based voting systems.

In 1984, Congress appropriated funds for the FEC to develop voluntary national Standards for computer-based voting systems. The FEC formally approved the *Performance and Test Standards for Punchcard, Optical Scan and Direct Recording Electronic Voting Systems*[1] in January 1990.

The national testing effort was developed and overseen by the National Association of State Election Director's (NASED's) Voting Systems Board, which is composed of election officials and independent technical advisors. NASED's testing program was initiated in 1994 and more than 30 voting systems or components of voting systems have gone through the NASED testing and qualification process. In addition, many systems have subsequently been certified at the state level using the Standards in conjunction with functional and technical requirements developed by state and local policymakers to address the specific needs of their jurisdictions.

As the qualification process matured and qualified systems were used in the field, the Voting Systems Board, in consultation with the testing labs, was able to identify certain testing issues that needed to be resolved. Moreover, rapid advancements in information and personal computer technologies introduced new voting system development and implementation scenarios not contemplated by the 1990 Standards.

In 1997, NASED briefed the FEC on the necessity for continued FEC involvement, citing the importance of keeping the Standards current in its reflection of modern and emerging technologies

[1] This document is generally referred to as the *Voting Systems Standards.*

employed by voting system vendors. Following a Requirements Analysis released in 1999, the Commission authorized the Office of Election Administration to revise the Standards to reflect contemporary needs of the elections community. This resulted in the 2002 Voting System Standards.

In 2002, Congress passed the Help America Vote Act, which created a new process for improving voluntary voting system guidelines. A new federal entity was created, the Election Assistance Commission, to oversee the process. The EAC established the Technical Guidelines Development Committee in accordance with the requirements of section 221 of HAVA pursuant to the Federal Advisory Committee Act, 5 U.S.C. App. 2. The objectives and duties were to act in the public interest to assist the EAC in the development of the voluntary voting system guidelines. The membership, as defined by HAVA, includes:

- The Director of the National Institute of Standards and Technology (NIST) who shall serve as its chair,
- Members of the Standards Board,
- Members of the Board of Advisors,
- Members of the Architectural and Transportation Barrier, and Compliance Board (Access Board),
- A representative of the American National Standards Institute,
- A representative of the IEEE,
- Two representatives of the NASED selected by such Association who are not members of the Standards Board or Board of Advisors, and who are not of the same political party, and
- Other individuals with technical and scientific expertise relating to voting systems and voting equipment.

The TGDC first met in August 2004 and delivered its initial set of recommendations to the EAC in April 2005. The initial set of recommendations augments the VSS 2002 by including security measures for auditability, wireless communications and software distribution and set up, and improvements for the accessibility guidelines and usability design guidelines. The TGDC also recommended that the VSS 2002 should be replaced with a far-reaching guideline that would address in-depth security, performance-based guidelines for usability testing, and an overhaul of the standards and test methods to meet today's more rigorous needs for electronic voting systems.

1.3 Issues Addressed by the VVSG Version 2 (April 2005)

VVSG Version 2 (April 2005) begins the process of reorganizing the 2002 VSS to better serve the needs of voting system and product developers, election officials, and testing laboratories. This initial draft offers initial requirements to ensure that the voting process shall accurately accumulate, count, and report legitimate votes. An overview of security requirements and a human factors roadmap are also included. A voting process model assists in defining terms used in the requirements process. The Appendices in this document include an analysis table of the 2002VSS requirements; revisions to Chapters 4 and 5 of the 2002 VSS; and a glossary.

1.4 Organization of VVSG Version 2

Eight organizational principles describe a high-level framework for defining the functions of a voting system. The functions map into a voting process model. In this initial version of the standard, NIST has generated standards development work on a sub set of the principles.

1. All eligible and potentially eligible voters shall be allowed access to the voting process without discrimination.

2. The voting process must ensure: (1) that each and every legitimate voter has exactly one ballot counted, and (2) that no other ballots are counted.

3. Each cast ballot shall capture the intent of the voter who cast that ballot.

4. The voting process shall accurately accumulate, count, and report legitimate votes.

5. The voting process shall preserve the secrecy of the ballot and not allow the voter or the voting process to reveal their votes.

6. Equipment and associated procedures shall be fit for the purpose of carrying out the voting process and shall be appropriate for use by voting officials.

7. The voting process shall be resilient to disruptions.

8. Independent observers shall be able to verify the correct operation of the voting process.

2 Guiding Principles

2.1 All eligible and potentially eligible voters shall be allowed access to the voting process without discrimination.

This requirement deals with the right of the voter to have equal access to the voting process, that is, to be able to enter the polling place and have access to voting equipment so as to cast a ballot. It is to ensure that the voting process does not discriminate against voters and prevent them from voting or impeding their ability to vote based on their language, accessibility requirements, or membership in certain subpopulations, e.g., economic, racial, political, religious.

This requirement contains sub-requirements dealing with the physical, environmental, usability, and accessibility conditions for voting and to support voting. The sub-requirements deal primarily with the voting registration process and access to voting precincts and equipment. It does not deal specifically with requirements for the usability of voting equipment, which is covered in requirement 3. This requirement is impacted by requirement 5, which requires that accommodations be made for voter privacy; requirement 1 implies that the usability and accessibility requirements of voters must be accommodated so that they may vote in private.

2.2 The voting process must ensure: (1) that each and every legitimate voter has exactly one ballot counted, and (2) that no other ballots are counted.

This requirement is to ensure that every legitimate voter's ballot is counted and that no other ballots can be inserted into the voting process and subsequently counted. Legitimate is used here to indicate that the voter has met the eligibility requirements for voting, has registered to vote in the election, and the voter's eligibility and registration have been verified by an election official prior to the counting of the ballots.

This requirement permits provisional voting, in which voters who may or may not be eligible to vote are able to cast ballots. This requirement also permits voting schemes such as those in which voters are able to vote many times but only their last vote is counted. This requirement does not permit the count of cast ballots to occur until all ballots have been ascertained to have been cast by legitimate voters.

This requirement must be modified to address the non-counting of provisional, absentee, or other votes. Currently, some elections are conducted in which these votes are not counted until possibly after the Election Day tallies are calculated, and they are counted only if it has been determined that the counting of the votes can affect the outcome of the election.

2.3 Each cast ballot shall capture the intent of the voter who cast that ballot.

This requirement deals with the right of the voter to have their ballot presented to them in a manner that reasonably accommodates their usability and accessibility requirements. This requirement also deals with the right of the voter to have their ballot captured accurately by the voting process and its equipment. This means that the voter's intent must be captured exactly by the voting process. Reducing overvotes and unintentional undervotes is covered by this requirement.

This requirement is impacted by requirement 4, which requires that the voting process notify the voter of mistakes or any other conditions of the voting process and equipment that may prevent the intent of the voter from being captured.

2.4 The voting process shall accurately accumulate, count, and report legitimate votes.

This requirement deals with the process of accurately accumulating or recording ballots cast by legitimate voters and subsequently counting them and reporting the final tallies of votes. The voting process, or more specifically the voting equipment, must be able to perform these functions to a high degree of accuracy, and produce evidence for verification of the correctness of the functions.

This requirement impacts requirement 3 in that it addresses errors that prevent cast ballots from being accurately accumulated. It also is impacted by requirement 8, which requires that the accumulation, count and reporting of votes be done in a manner that is auditable and verifiable.

2.5 The voting process shall preserve the secrecy of the ballot and not allow the voter or the voting process to reveal their votes.

This requirement deals with two central issues regarding voter privacy:

- the voting system or the way it is used must not reveal how a voter cast their ballot, and
- the voting system must minimize the ability of a voter to prove to others how they voted, e.g., not print a copy of the voter's ballot for the voter's own use.

This requirement includes sub-requirements to prevent voters' choices from being made known by the voting process. For example, voting systems must randomize ballots as they are cast and take other appropriate measures so that attempts to determine the intent of voters based on the order of cast ballots, including ballots cast by various subpopulations, will fail.

Some cryptographic-based voting protocols permit the voter to leave the polling place with a cryptographic checksum of the voter's ballot or a partial representation of the ballot that is insufficient for use in proving how one voted; such protocols do not violate the intent of this requirement.

In situations where absentee ballots are used or in certain remote voting situations in which voter privacy is not possible, voters may easily show their ballots to others; therefore this requirement must be balanced with requirement 1.

2.6 Equipment and associated procedures shall be fit for the purpose of carrying out the voting process and shall be appropriate for use by voting officials.

This requirement deals with the physical characteristics of physical equipment used in the voting process and the procedures used for management of the equipment. Therefore, it deals with issues such as size and weight of equipment, as well as materials construction and electrical components. It also deals with the usability of the systems with regard to election management and administration as well as whether the procedures are suited for operation by election officials and

other voting system staff. For example, this requirement mandates that the procedures for robust operation of equipment accommodate the typical abilities of voting officials.

This requirement does not deal with or overlap other requirements related to correctness of voting systems or voter usability requirements. For example, issues related to the proper handling of cast ballots are covered in requirement 4. Issues related to ballot usability are covered in requirement 3.

2.7 The voting process shall be resilient to disruptions.

This requirement is concerned with the ability of the voting process and voting equipment to withstand disruptions that may result from many factors, including the following:

- natural disasters,
- disruptions of electrical service, or
- electronic or physical denial of service attacks.

Therefore, the voting process must take into account the environment in which it takes place and must incorporate appropriate contingencies, security mechanisms, and other measures so that the voting process is not disrupted. For example, the voting process must be resilient to certain acts of nature as they are more likely to occur in certain areas, e.g., snow in northern climates. If, for example, voting systems are computer-based, they must be resilient to electronic attacks and other attacks such as computer viruses. Additionally, the voting process must be resilient to voter-based attacks, such as deliberate or accidental spoilage of ballots.

2.8 Independent observers shall be able to verify the correct operation of the voting process.

This requirement deals with verifying the correct operation of the voting process and voting equipment. Because requirement 5 mandates that the voting process shall preserve the secrecy of the ballot, requirement 8 is made more difficult in that verification cannot use audit logs or receipts to conclusively show how individual voters cast their ballots. The correct operation, therefore, must be verified at various steps during the voting process, including the following:

- during development of the voting system in which verification can occur through better system and software design and independent recording of votes,
- during pre-election testing in which verification can occur through robust testing of voting equipment and procedures,
- during election day operation of the voting system in which verification can occur through random inspections and examination of audit records, and
- during post-election procedures in which verification can occur through examination of audit records and independent records of votes.

This requirement deals with trade-offs that may occur depending on voting system design. For example, electronic voting systems whose security-related functions are closely coupled with user

interfaces or commercial off-the-shelf software (COTS) must undergo inspections that may be different from those for voting systems that decouple those functions or that produce independent records of votes.

3 Voting Process Model

3.1 Catalog of Activities and Objects

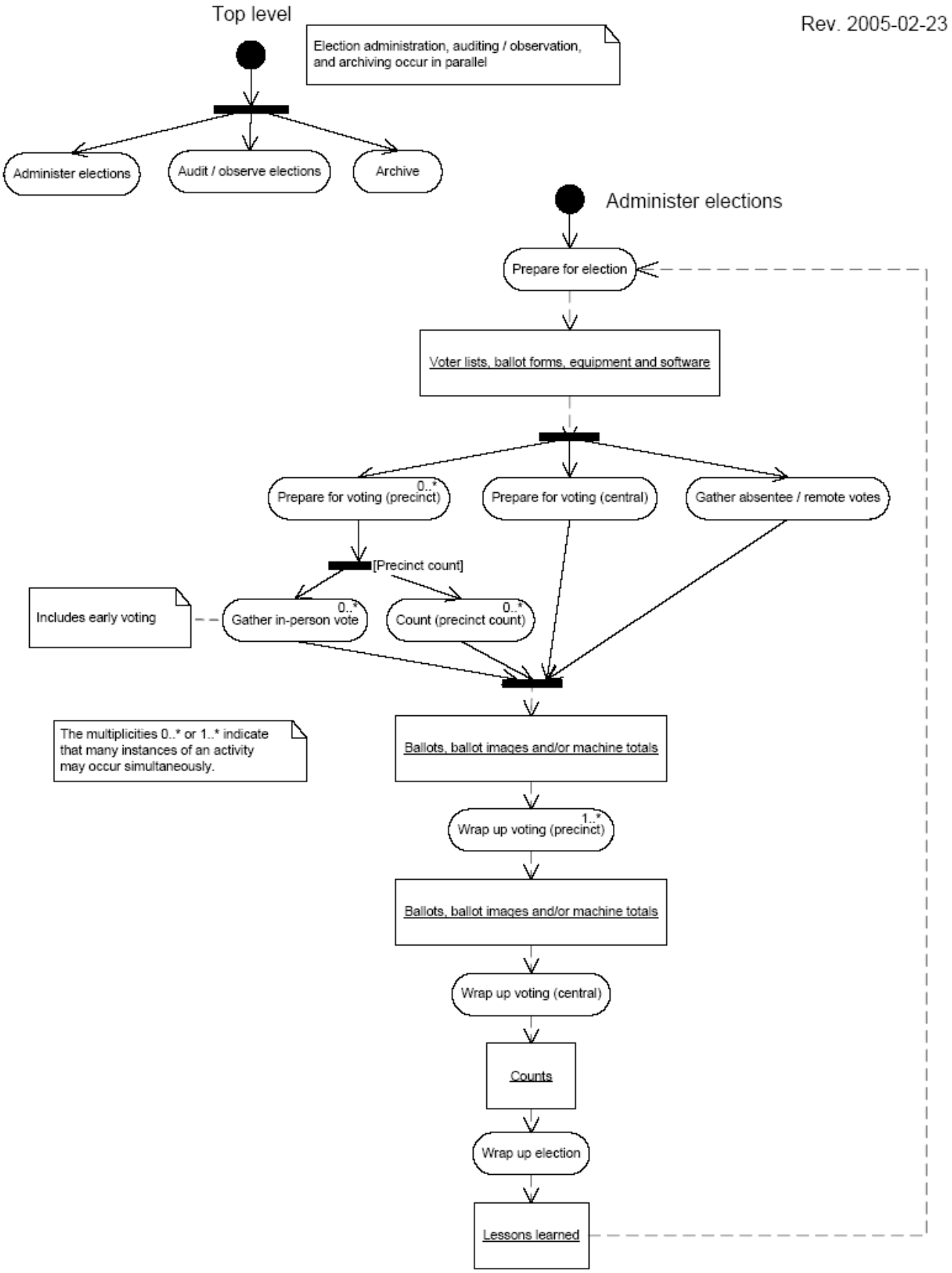
The following is an enumeration and identification of the activities and objects that appear in the voting process model diagrams.

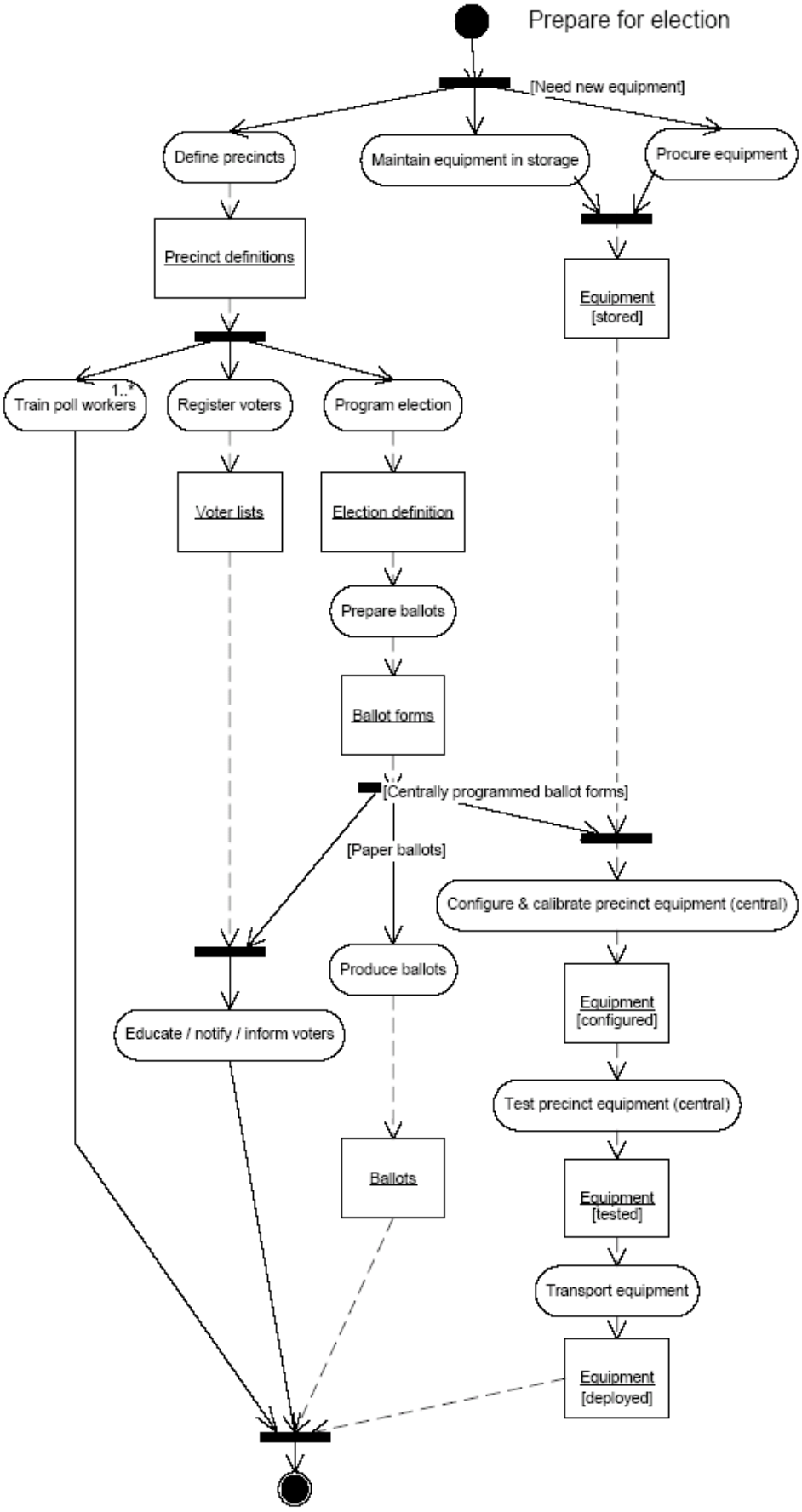
Activities are assigned identifiers of the form AX.Y.Z, where the prefix A indicates that it is an activity and the outline numbering X.Y.Z indicates the nesting of sub activities within activities. This follows the structure of the diagrams.

In some cases, a given activity appears in two variants, first as a precinct activity and then as a central activity. The precinct and central variants are assigned different numbers to enable discussion of the different concerns for precinct versus central count systems. However, many requirements will apply equally to both variants, in which case both numbers will be referenced.

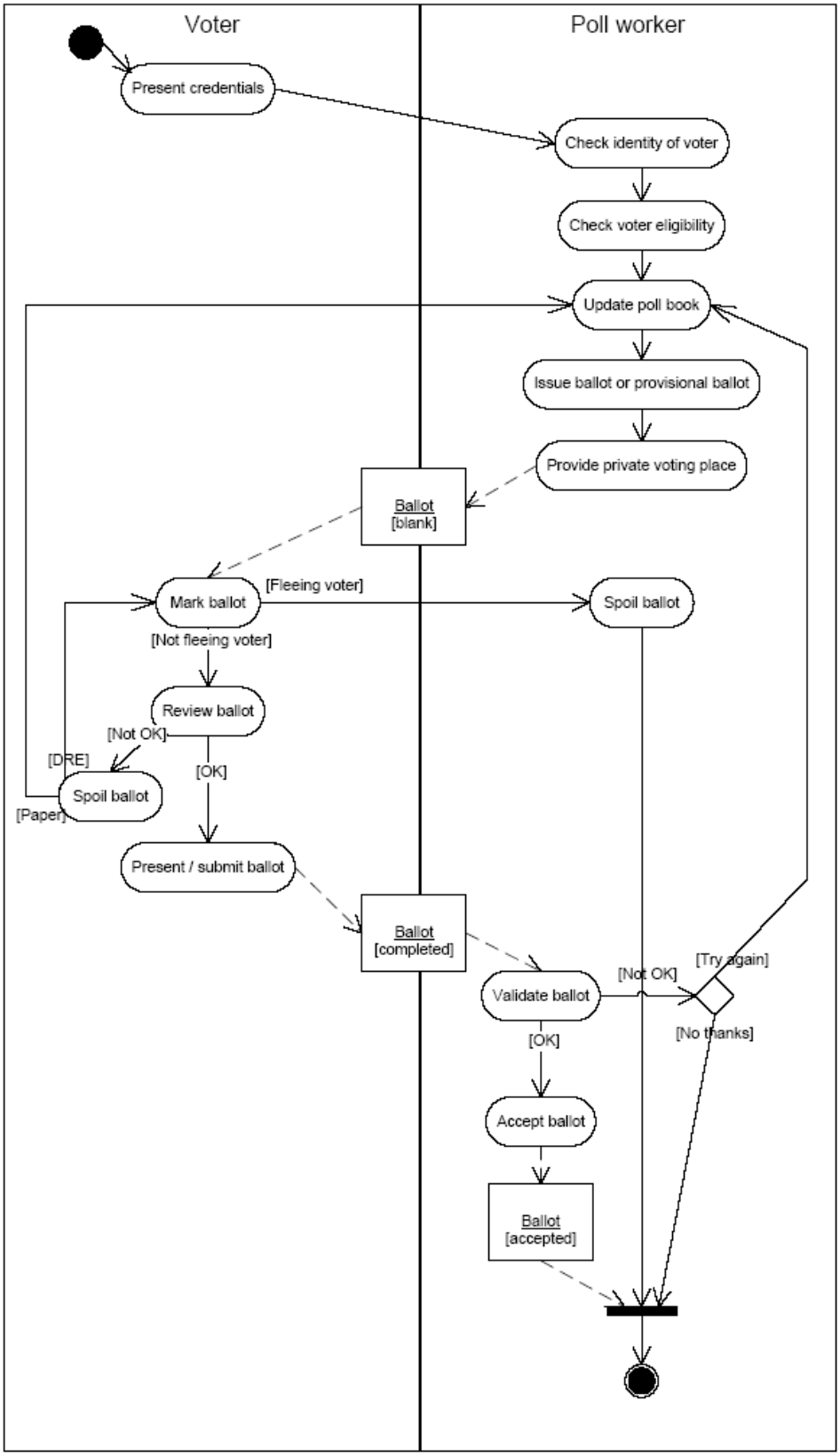
Objects are assigned identifiers of the form OX, where the prefix O indicates that it is an object and the number X is simply a serial number. The same object may appear in multiple activities and in multiple roles within an activity. These different uses of the same object are not assigned different numbers.

Notations in all of the diagrams provide information about the sequencing of and dependencies among activities and objects. This information is translated into text in Section 2.

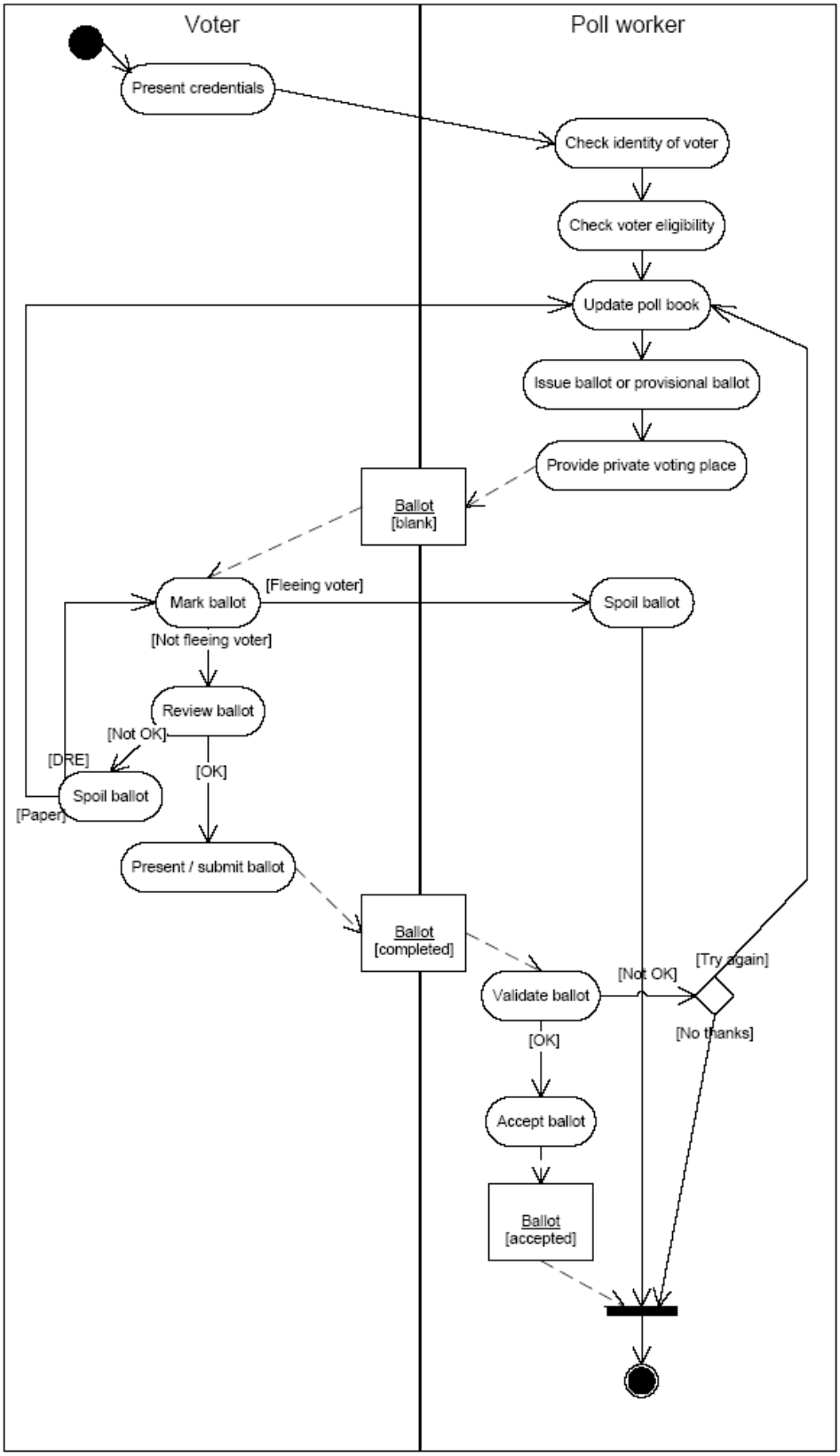

Top level
Rev. 2005-02-23
Election administration, auditing / observation, and archiving occur in parallel
Administer elections
Audit / observe elections
Archive
Administer elections
Prepare for election
Voter lists, ballot forms, equipment and software
Prepare for voting (precinct)
0..*
Prepare for voting (central)
Gather absentee / remote votes
[Precinct count]
Includes early voting
Gather in-person vote
0..*
Count (precinct count)
0..*
The multiplicities 0..* or 1..* indicate that many instances of an activity may occur simultaneously.
Ballots, ballot images and/or machine totals
Wrap up voting (precinct)
1..*
Ballots, ballot images and/or machine totals
Wrap up voting (central)
Counts
Wrap up election
Lessons learned

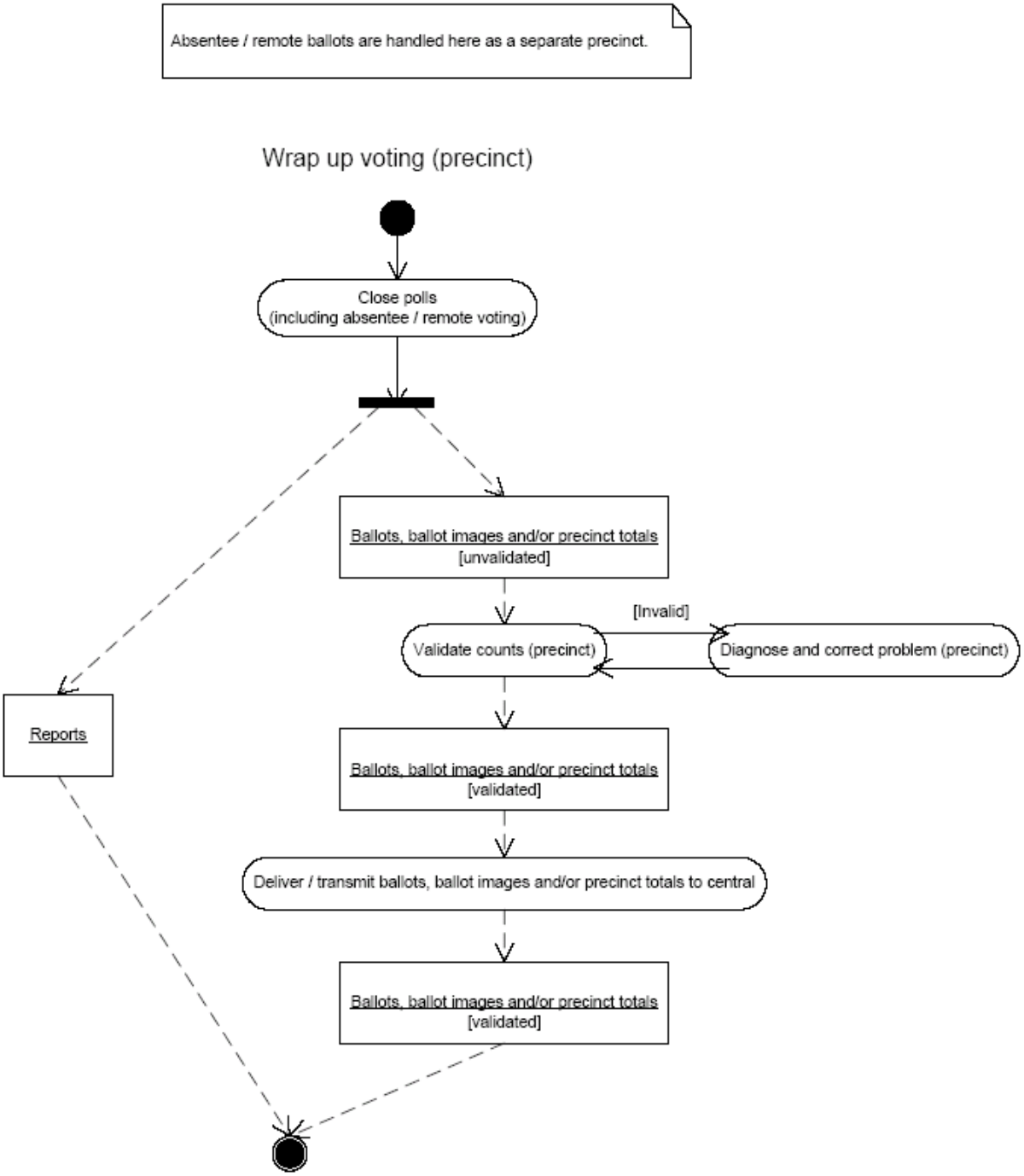

Prepare for election
[Need new equipment]
Define precincts
Maintain equipment in storage
Procure equipment
Precinct definitions
Equipment [stored]
Train poll workers
1..*
Register voters
Program election
Voter lists
Election definition
Prepare ballots
Ballot forms
[Centrally programmed ballot forms]
[Paper ballots]
Configure & calibrate precinct equipment (central)
Produce ballots
Equipment [configured]
Educate / notify / inform voters
Test precinct equipment (central)
Ballots
Equipment [tested]
Transport equipment
Equipment [deployed]

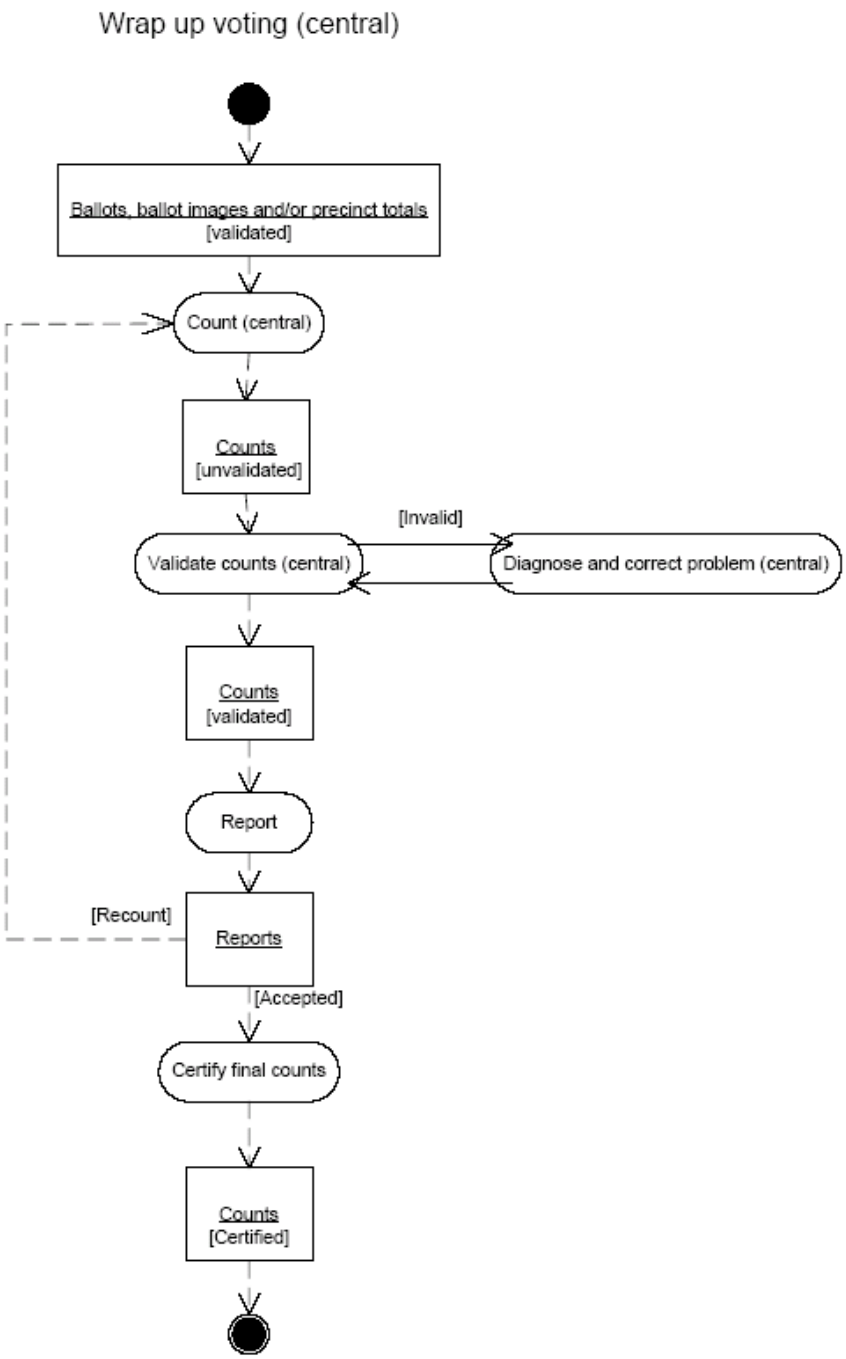
Gather in-person vote

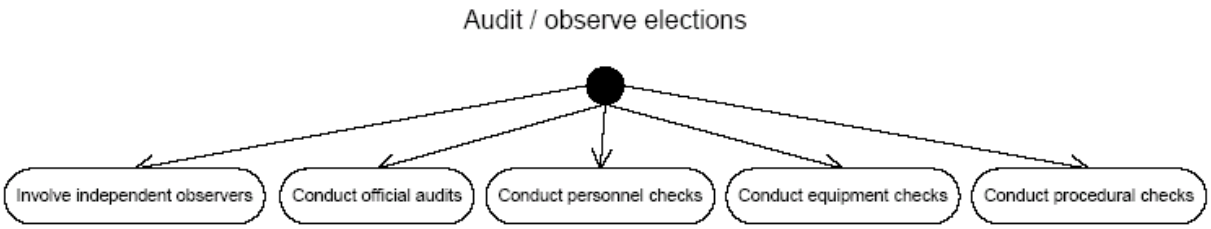


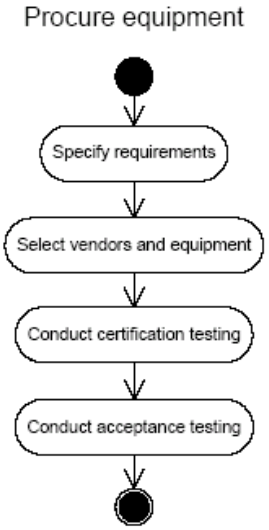
Gather in-person vote

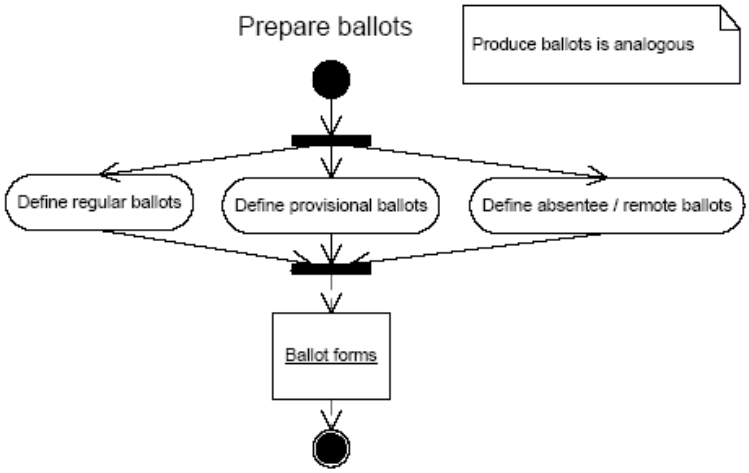


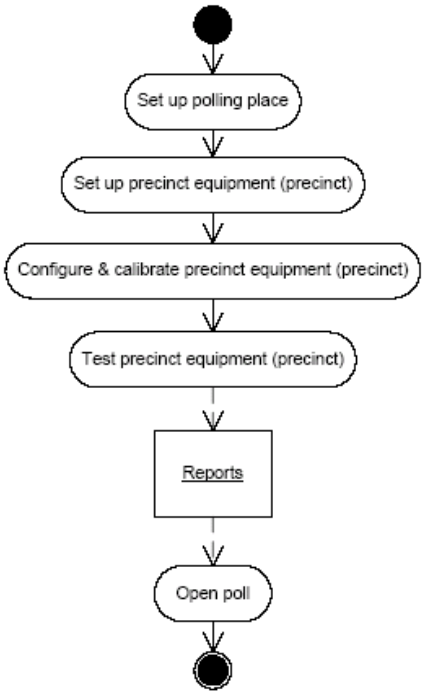

Absentee / remote ballots are handled here as a separate precinct.
Wrap up voting (precinct)
Close polls
(including absentee / remote voting)
Ballots, ballot images and/or precinct totals
[unvalidated]
[Invalid]
Validate counts (precinct)
Diagnose and correct problem (precinct)
Reports
Ballots, ballot images and/or precinct totals
[validated]
Deliver / transmit ballots, ballot images and/or precinct totals to central
Ballots, ballot images and/or precinct totals
[validated]

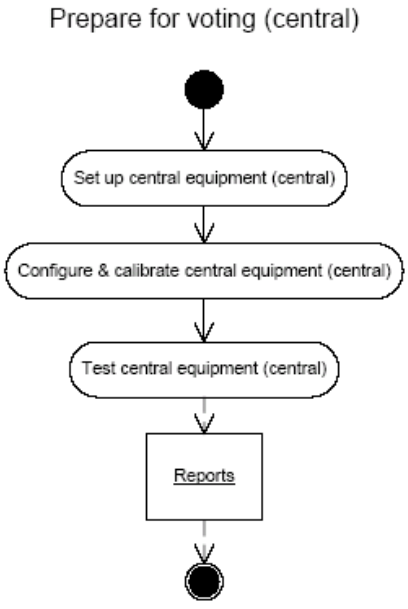

Wrap up voting (central)
Ballots, ballot images and/or precinct totals
[validated]
Count (central)
Counts
[unvalidated]
[Invalid]
Validate counts (central)
Diagnose and correct problem (central)
Counts
[validated]
Report
[Recount]
Reports
[Accepted]
Certify final counts
Counts
[Certified]


Audit / observe elections
Involve independent observers
Conduct official audits
Conduct personnel checks
Conduct equipment checks
Conduct procedural checks


Procure equipment
Specify requirements
Select vendors and equipment
Conduct certification testing
Conduct acceptance testing


Prepare ballots
Produce ballots is analogous
Define regular ballots
Define provisional ballots
Define absentee / remote ballots
Ballot forms


Prepare for voting (precinct)
Set up polling place
Set up precinct equipment (precinct)
Configure & calibrate precinct equipment (precinct)
Test precinct equipment (precinct)
Reports
Open poll


Prepare for voting (central)
Set up central equipment (central)
Configure & calibrate central equipment (central)
Test central equipment (central)
Reports

(Note: A notational convention is used to designate which party, Voter or Poll worker, is the actor for each activity in A1.5. Spoil ballot can be done by either party. Present credentials, Mark ballot, Review ballot, and Present / submit ballot are only done by Voter. All others are only done by Poll worker.)

(Note: Absentee / remote ballots are handled here as a separate precinct.)

3.2 Translation of Diagrams

Activities, which are represented in the diagrams by ovals, are represented in this translation by the activity name in parenthesis. Objects, which are represented in the diagrams by boxes, are represented in this translation by the object name in square brackets.

Sometimes the names of activities and objects will themselves be qualified by parenthetical phrases or object states in square brackets. These have been retained as-is, nesting the parenthesis or brackets as needed.

The control and object flow information in the diagrams has been translated into pseudo-code. The following keywords are used to indicate how flows occur.

Begin/End: Activities in the scope of Begin/End occur serially, i.e., one after the other, in the order listed.

ParBegin/ParEnd: Activities in the scope of ParBegin/ParEnd occur in parallel, i.e., they all occur simultaneously.

In some cases, control flows into a given activity only if a guard condition is met. This is represented by including the text of the guard, followed by a semicolon, before the affected activity.-- In some cases, control flows back to an earlier activity (a loop) or across to a different thread running in parallel (a synchronization point). When this occurs, the target of the flow is followed by an ellipsis, suggestive of the fact that the repetition of a previously detailed sequence has been omitted.

Translation of the diagrams follows.

Diagram: Top level

ParBegin
(Administer elections)
(Audit / observe elections)
(Archive)
ParEnd

Diagram: Administer elections

```
Begin
  (Prepare for election)
  [Voter lists, ballot forms, equipment and software]
  ParBegin
    Begin
      (Prepare for voting (precinct))
      ParBegin
          (Gather in-person vote)
          Precinct count:  (Count (precinct count))
      ParEnd
    End
    (Prepare for voting (central))
    (Gather absentee / remote votes)
  ParEnd
  [Ballots, ballot images and/or machine totals]
  (Wrap up voting (precinct))
  [Ballots, ballot images and/or machine totals]
  (Wrap up voting (central))
  [Counts]
  (Wrap up election)
  [Lessons learned]
  (Prepare for election)...
End
```

Diagram: Prepare for election

```
ParBegin
  Begin
    ParBegin
      (Maintain equipment in storage)
      Need new equipment:  (Procure equipment)
    ParEnd
    [Equipment [stored]]
    (Configure & calibrate precinct equipment (central))
    [Equipment [configured]]
    (Test precinct equipment (central))
    [Equipment [tested]]
    (Transport equipment)
    [Equipment [deployed]]
  End
  Begin
    (Define precincts)
    [Precinct definitions]
    ParBegin
```

```
    (Train poll workers)
    Begin
      (Register voters)
      [Voter lists]
      (Educate / notify / inform voters)
    End
    Begin
      (Program election)
      [Election definition]
      (Prepare ballots)
      [Ballot forms]
      ParBegin
        (Educate / notify / inform voters)...
        Centrally programmed ballot forms:  (Configure & calibrate precinct equipment (central))...
        Paper ballots:  Begin
          (Produce ballots)
          [Ballots]
        End
      ParEnd
    End
  ParEnd
 End
ParEnd
```

Diagram: Gather in-person vote

Note: A notational convention is used to designate which party, Voter or Poll worker, is the actor for each activity in A1.5. Spoil ballot can be done by either party. Present credentials, Mark ballot, Review ballot, and Present / submit ballot are only done by Voter. All others are only done by Poll worker.

```
Begin
 (Present credentials)
 (Check identity of voter)
 (Check voter eligibility)
 (Update poll book)
 (Issue ballot or provisional ballot)
 (Provide private voting place)
 [Ballot [blank]]
 (Mark ballot)
 ParBegin
   Fleeing voter:  (Spoil ballot)
   Not fleeing voter:  Begin
     (Review ballot)
     ParBegin
          Not OK:  Begin
```

```
          (Spoil ballot)
          ParBegin
            DRE:  (Mark ballot)...
            Paper:  (Update poll book)...
          ParEnd
        End
        OK:  Begin
          (Present / submit ballot)
          [Ballot [completed]]
          (Validate ballot)
          ParBegin
            OK:  Begin
              (Accept ballot)
              [Ballot [accepted]]
            End
            Not OK: Try again:  (Update poll book)...
          ParEnd
        End
    ParEnd
  End
 ParEnd
End
```

Diagram: Wrap up voting (precinct)

```
Begin
 (Close polls (including absentee / remote voting))
 ParBegin
  [Reports]
  Begin
   [Ballots, ballot images and/or precinct totals [unvalidated]]
   (Validate counts (precinct))
   ParBegin
       Invalid:  Begin
         (Diagnose and correct problem (precinct))
         (Validate counts (precinct))...
       End
       Valid:  Begin
         [Ballots, ballot images and/or precinct totals [validated]]
         (Deliver / transmit ballots, ballot images and/or precinct totals to central)
         [Ballots, ballot images and/or precinct totals [validated]]
       End
   ParEnd
  End
 ParEnd
End
```

Diagram: Wrap up voting (central)

```
Begin
  [Ballots, ballot images and/or precinct totals [validated]]
  (Count (central))
  [Counts [unvalidated]]
  (Validate counts (central))
  ParBegin
    Invalid:  Begin
      (Diagnose and correct problem (central))
      (Validate counts (central))...
    End
    Valid:  Begin
      [Counts [validated]]
      (Report)
      [Reports]
      ParBegin
          Recount:  (Count (central))...
          Accepted:  Begin
            (Certify final counts)
            [Counts [certified]]
          End
      ParEnd
    End
  ParEnd
End
```

Diagram: Audit / observe elections

```
ParBegin
  (Involve independent observers)
  (Conduct official audits)
  (Conduct personnel checks)
  (Conduct equipment checks)
  (Conduct procedural checks)
ParEnd
```

Diagram: Procure equipment

```
Begin
  (Specify requirements)
  (Select vendors and equipment)
  (Conduct certification testing)
  (Conduct acceptance testing)
End
```

Diagram: Prepare ballots
(Note: Produce ballots is analogous.)

Begin
ParBegin
(Define regular ballots)
(Define provisional ballots)
(Define absentee / remote ballots)
ParEnd
[Ballot forms]
End

Diagram: Prepare for voting (precinct)

Begin
(Set up polling place)
(Set up precinct equipment (precinct))
(Configure & calibrate precinct equipment (precinct))
(Test precinct equipment (precinct))
[Reports]
(Open poll)
End

Diagram: Prepare for voting (central)

Begin
(Set up central equipment (central))
(Configure & calibrate central equipment (central))
(Test central equipment (central))
[Reports]
End

Diagram: Register voters

Begin
[Registration database [original]]
ParBegin
(Register new voters)
(Update voter information)
(Purge ineligible, inactive, or dead voters)
ParEnd
[Registration database [updated]]
(Generate voter lists)
[Voter lists]
End

Diagram: Wrap up election

ParBegin
(Deactivate equipment)
(Conduct post-mortem)
ParEnd

Diagram: Deactivate equipment

Begin
(Pack up equipment)
(Transport equipment)
(Put equipment in storage)
End

Diagram: Conduct post-mortem

Begin
(Analyze election results)
[Lessons learned]
(Refine needs and requirements)
(Make revisions / changes to existing hardware, software, processes, procedures, and testing)
End

4 Requirements

4.1 Requirements for Principle 2.3

4.1.1 Introduction

This section describes a roadmap for the area of human factors and privacy that applies, for the most part, to the goal of capturing the intent of the voter, that is, the indication of a voter's choice. It is motivated by the TGDC Resolutions #2-05 through #12-5 that were approved in January 2005. This roadmap will form the basis of the human factors and privacy work planned for the development of the Voluntary Voting System Guidelines (VVSG), Version 2. We do not discuss, in any detail, the roadmap beyond VVSG2, in this document.

4.1.2 Accessible Voting System Requirements

The goal is to improve the requirements for accessibility. Based on the feedback we receive on the accessibility requirements in VVSG1, we will improve the guidelines by correcting any errors found in the VVSG1, and extending the guidelines to better capture requirements for next

generation systems. These will include some draft performance benchmarks for voter speed and accuracy and usability test protocols for different types of disabilities.

4.1.3 Human Factors and Privacy at the Polling Place

The goal is to improve the requirements for human factors and privacy at the polling place. VVSG1 requires that the polling places conform to the appropriate guidelines of the Americans with Disabilities Act (ADA) of 1990 and of the Architectural Barriers Act (ABA) of 1968. There are also privacy requirements for configuring the polling place and voting station so as to prevent others from learning the contents of a voter's ballot.

For the VVSG2, we plan to do a more thorough analysis to create requirements that are more specific to voting. For example, The US Department of Justice has a "ADA Checklist for Polling Places" http://www.ada.gov/votingck.htm and material from the Design for the Democracy project, http://designfordemocracy.aiga.org, which contains design guidance for signage will be considered for possible integration into VVSG2 guidance for election officials and polling place workers.

4.1.4 Capturing Indication of a Voter's Choice

The goal is to improve requirements for human factors and privacy for voting systems for capturing the indication of a voter's choice. VVSG1 includes basic usability requirements. For VVSG2, we will develop performance-based requirements. The approach relies on the usability benchmark development described below.

4.1.4.1 Human Performance-Based Standards and Usability Testing

The goal is to perform research to develop performance requirements and conformance tests for usability.

We first define the performance measures. Usability is defined generally as a measure of the effectiveness, efficiency, and satisfaction achieved by a specified set of users with a given product in the performance of specified tasks. In the context of voting, the primary users are the voters (but also poll workers), the product is the voting system, and the task is the correct representation of one's choices in the election.

Additional requirements for task performance are independence and privacy: the voter should normally be able to complete the voting task without assistance from others (although the voting system itself may offer help), and the voter's choices should be private. Privacy in this context, including the property of the voter being unable to disclose his or her vote, ensures that the voter can make choices based solely on his or her own preferences without intimidation or inhibition.

Aside from its intrinsic undesirability, lack of independence or privacy may adversely affect effectiveness (e.g. by possibly inhibiting voters' ability to vote according to their own preference) and efficiency (e.g. by slowing down the process).

It is the intention of the TGDC that in forthcoming versions of the VVSG, usability will be addressed by high-level performance-based requirements. That is, the requirements will directly address metrics for

1. Effectiveness -- especially low error rate for marking the ballot; the voter's intention must be correctly conveyed to and represented within the voting system. NIST will propose precise metrics for the error rate.

2. Efficiency -- time and other resources taken to vote. Probably the easiest aspect to measure directly, since it involves merely timing test subjects.

3. Satisfaction -- voter experience is safe, comfortable, free of stress, and instills confidence. There are various survey instruments, such as SUMI or QUIS, which should provide the basis for measuring satisfaction.

Also see: Whitney Quesenbery et al, "Defining a Summative Usability Test for Voting Systems", Usability Professions Association, September 2004, http://www.upassoc.org/upa_projects/voting_and_usability/documents/voting_summative_test.pdf for a very thorough discussion of the issues for the design of good summative tests. In particular, note the need to define test ballots of varying complexity and the need to resolve demographic issues for selection of test subjects.
Once the metrics are defined, research needs to be done to support the development of human performance-based standards and associated usability testing for voting systems.
We note: "Although usability testing is widely employed as part of a user-centered design process, there is *little industry experience* in usability testing as part of the *certification of a system* [emphasis added]." ; from the UPA 2004 Workshop on Voting Systems.

There are 2 interdependent goals:

1. High quality performance *standards* for voting:
   -- Objective, measurable criteria
   -- Metrics directly address "bottom line" performance of equipment
   -- Fair to all technologies
   -- Criteria push technology improvement, yet are realistic

2. High quality performance *tests* for voting:
   -- Repeatable, reliable, valid
   -- Uncover even low-incidence errors
   -- Minimize technical complexity, burden on operator

Note that the goals are not targeted towards simple functional testing (e.g., does system support party-line voting?) as usability testing is not needed for this.

The steps needed to support the goals (some of these are already underway):

1. Obtain examples of currently used voting equipment - as wide ranging as possible; must include mechanism for generating test ballots. These will serve as the basic "laboratory equipment" for usability experimentation. (Underway)

2. Obtain a wide-ranging sample of recent ballots. (Underway)

3. Formulate preliminary metrics for effectiveness, efficiency, and satisfaction. In particular for effectiveness: analyze how to define errors (wrong candidate, failure to cast ballot, request for "poll worker" assistance) and error rates. For efficiency: how to aggregate timing data? Mean, median, or other statistic?

4. Formulate three test ballots of varying complexity: low, medium, high (based on #2).

5. Formulate preliminary moderator script, other instructions to test subjects, and other operational procedures. Note that these should all be technology-independent. For a better-controlled experiment, we will likely take the approach of telling subjects how to vote, rather than letting them decide and then report.

6. Outstanding issue: what is correct level of abstraction for instructions?

    - Result only? e.g. "vote for Jones" (tests usability of system as a whole for achieving final result).

    - Force exercising of some functional capabilities? "Vote straight party ticket" or "First vote Jones, then switch to Smith". This involves telling subject not only what to accomplish, but to some extent how (tests usability of certain sub-systems for achieving result in a certain way). Or is this just a "sub-metric" whose effect is captured in the overall effectiveness metric?

7. Decide how voter sessions are to be recorded: note especially the level on monitoring to be done (use video?) and automatic capture (as much as possible) of error data and timing data. May involve extra instrumentation of voting equipment.

8. Preliminary design of satisfaction questionnaire to be used. Get necessary Paperwork Reduction Act permission.

9. Get necessary permission for human subjects and recruit manageable number of initial test subjects. Run them through the preliminary protocol. The purpose here is not so much to gather any reliable usability data as to see which parts of the protocol work smoothly and which need refinement.

10. Based on initial experiments and iteration, refine elements from #3-8 above. Milestone: some confidence in the basic protocol has been established.

11. Based on initial experiments, formulate preliminary performance benchmarks for effectiveness, efficiency, and satisfaction.

12. Formulate preliminary statistical approach: based on #11, what level of error rate do we aim to measure, how many subjects needed, what confidence levels, use of Wald formula, etc.

13. Formulate approach for demographic selection of subjects. How to choose demographics? What characteristics are relevant (education, age, familiarity with technology)?

14. Recruit number of subjects sufficient to validate statistical and demographic approach.

15. Run "full-scale" usability tests. Re-validate basic protocol. Validate statistical approach:

    - Reproducibility of results per voting system (Given similar subjects, System A always scores about the same.)

    - Reproducibility across voting systems, given similar subjects.(System A is always 30% faster than system B).

    - Reproducibility and significance of demographic effects (e.g., older voters are more accurate, but slower).

16. Propose final protocols and benchmarks.

Sources: HFP report, UPA report

-- "NIST Special Publication 500-256, Improving the Usability and Accessibility of Voting Systems and Products". See: http://vote.nist.gov/Final%20Human%20Factors%20Report%20%205-04.pdf

-- Whitney Quesenbery et al, "Defining a Summative Usability Test for Voting Systems", Usability Professions Association, September 2004. See: http://www.upassoc.org/upa_projects/voting_and_usability/documents/voting_summative_test.pdf

4.1.4.2 Accommodating a Wide Range of Human Abilities

The goal is to investigate existing universal usability and design guidelines and apply them to create a set of voting system guidelines and standards.

In the course of developing the VVSG1 , we noted that some of what might be called “accessibility” requirements apply to all electronic voting stations rather that just the accessible

voting stations. Examples of these include requirements for default color, screen flicker, and adjustable font size. We also note that some accessibility requirements support multiple disabilities. For example, the synchronization of the audio ballot with video output accommodates a much wider range of people with visual disabilities than the audio ballot alone, as well as accommodating those with dyslexia. We will analyze the literature of universal design to see what other types of basic requirements could be easily implemented in all voting stations. For example, The Center for Universal Design at North Carolina State University has identified 7 Principles of Universal Design. See http://www.design.ncsu.edu:8120/cud/univ_design/princ_overview.htm . They define universal design as "the design of products and environments to be usable by all people, to the greatest extent possible, without the need for adaptation or specialized design."
From this, we think it is possible to create a small set of universal design requirements for voting systems.

4.1.4.3 Usability Guidance for Instructions, Ballot Design, and Error Messages

The goal is to review existing guidance for instructions, ballot design and error messages and draft new guidance as required, including incorporating "plain language" and form layout research. We will gather existing ballot design guidance and also consult with experts in these areas.

4.1.4.4 General Voting System Human Factors and Privacy Considerations

The goal is to develop improved requirements for human factors and privacy for voting systems where there is voter or election official interaction, other than for capturing the indication of a voter's choice. We have begun this work with the identification of requirements for voting officials and poll workers in the VVSG1. Next steps include looking at how to put these requirements into usable, how-to guidance.

4.1.4.4.1 Usability of the Standards

The goal is to evaluate the proposed new standards for usability. This work will begin when we have a draft of the VVSG2.

4.1.4.4.2 Availability of Voting Machines for Validating Benchmarks and Conformance Test Protocols

We have called for vendors to lend voting equipment to NIST via a Federal Register notice. We have identified space and support staff for the equipment. The next step is to arrange for vendors to give us instructions on how to setup their machines, including ballot creation.

4.2 Requirements for Principle 2.4

4.2.1 Rationale

4.2.1.1 Preface

This document contains rationale and discussion related to the scope of work of the Core Requirements and Testing Subcommittee as of 2005-03-29. Other, very important discussion for security, usability, accessibility, and fitness for purpose is not contained here but is or will be distributed separately.

4.2.1.1.2 Strategy

4.2.1.1.2.1 General

####### 4.2.1.1.2.1.1 Review of existing standards, specifications, and related work

(To ensure that previously written requirements would not be overlooked, NIST reviewed the resources listed in 4.2.1.1.2.1.2)

The resulting guide to existing requirements has not been put into publishable form but is being utilized by project members as they develop new recommendations.

NIST also reviewed sample ballot forms, vote data reports and other materials from several states.

####### 4.2.1.1.2.1.2 Standards, draft standards, regulations, and guidelines

[HAVA] Help America Vote Act of 2002, Public Law 107-252, 2002-10-29.
[2002VSS] 2002 Voting Systems Standards, available from http://www.eac.gov/election_resources/vss.html.

[P1583/D5.3.1] IEEE Draft Standard for the Evaluation of Voting Equipment, draft 5.3.1, 2004-10-08, available from http://grouper.ieee.org/groups/scc38/1583/private/ (password-protected).

[CoE 2004-09-30] Council of Europe, Committee of Ministers to member states on legal, operational, and technical standards for e-voting, adopted by the Committee of Ministers on 2004-09-30 at the 898th meeting of the Minister's Deputies, e-mail from Lori Steele, 2004-11-10.

[EML3] Election Markup Language v3.0, 2003-02-24, available from http://www.oasis-open.org/committees/election/index.shtml.

[SP 500-256] Sharon J. Laskowski et al., "Improving the Usability and Accessibility of Voting Systems and Products," NIST SP 500-256, 2004-05.

[508] Section 508 of the Rehabilitation Act: Electronic and Information Technology Accessibility Standards, 2000-12-21, available from http://www.access-board.gov/508.htm.

[ADA] ADA Checklist for Polling Places, 2004-02, available from http://www.usdoj.gov/crt/ada/votingchecklist.htm.
Issue lists

[D5.3.1 Comments 2004-10-19] Comments for d5-3-1 dated 10-19-2004 revC.xls, available from http://grouper.ieee.org/groups/scc38/1583/private/ (password-protected).

[5.0 Software Comments 2004-09-01] Software comments 5.0 (9-01-04).xls, available from http://grouper.ieee.org/groups/scc38/1583/private/ (password-protected).

[5.0 Security Comments 2004-08-18] Security extract V5 Comments – 2nd NJ Meeting.xls, available from http://grouper.ieee.org/groups/scc38/1583/private/ (password-protected).

[5.0 Reliability Accuracy Comments 2004-09-06] 5.0 Comments Section 5.2 & 6.2 (9-6-04).xls, available from http://grouper.ieee.org/groups/scc38/1583/private/ (password-protected).

[5.0 Accessibility Comments 2004-08-01] V5 Ballot Accessibility Comments – TG3 (8-1-04) .xls, available from http://grouper.ieee.org/groups/scc38/1583/private/ (password-protected).

[5.0 Environmental 2004-08-15] 5.0 Comments Section 5.4 & 6.4.xls, available from http://grouper.ieee.org/groups/scc38/1583/private (password-protected).

[5.0 EMC 2004-08-23] 5.0 Comments Section 5.5 (8-23-04).xls, available from http://grouper.ieee.org/groups/scc38/1583/private/ (password-protected).

[5.0 Provisional 2004-09-10] Gough-Provisional Ballot Comments.xls, available from http://grouper.ieee.org/groups/scc38/1583/private/ (password-protected).

[5.0 COTS 2004-06-18] Resolutions for COTS Comments for Draft 5.0 of IEEE P-1583, http://www.lipsio.com/COTS/docs/COTS.resolved.html.

[5.0 TDP 2004-04-23] 5.0 p1583 _TDP-Proposed resolution_Apr04.xls, available from http://grouper.ieee.org/groups/scc38/1583/private/ (password-protected).

[5.0 Comments 2003-10-16] Ballot Comment Form 5-0 10-16-2003.xls, available from http://grouper.ieee.org/groups/scc38/1583/private/ (password-protected).

4.2.1.1.2.1.3 Requests for proposals

[AZ] "OCR and DRE Voting Equipment – Statewide," Request for Proposal, Arizona, 2003. E-mail from Allan Eustis, 2004-10-12.

[CO-REG] "Statewide Voter Registration System," Request for Proposals # DOS-HAVA-0001, Colorado, 2004-01-16, formerly available from htt;://www.sos.state.co.us/pubs/hava/have_main.htm (now gone).

[CO-IVV] "Independent Verification and Validation for SCORE Project," Request for Proposals # DOS-HAVA-0002, Colorado, 2004-06-03, formerly available from http://www.sos.state.co.us/pubs/hava/hava_main.htm (now gone).

[GA] Request for Proposal GTA000040, Georgia, 2001. E-mail from Merle King via Allan Eustis, 2004-10-11.

[MD] "Direct Recording Electronic Voting System and Optical Scan Absentee Voting System for Four Counties," Project Number SBE-2002-01, Maryland, 2001-07-17, available from http://www.elections.state.md.us/citizens/voting_systems/voting_system_procurement.html.

[MI] Invitation To Bid # 071I4001011, Michigan, 2003, available from http://www.michigan.gov/sos/0,1607,7-127-1633_11619_27151-77943--,00.html.

[OH-VOT] "Statewide Voting System(s)," Request For Proposal # SOS0428365, Ohio, 2003-05-23, available from http://www.sos.state.oh.us/sos/hava/index.html.

[OH-REG] Request For Proposal # SOS032786279, Ohio, 2003-04-09, available from http://www.sos.state.oh.us/sos/hava/index.html.

[UT] "Executive Summary: Voting Equipment Selection Committee Request for Proposal," Utah. E-mail from Allan Eustis, 2004-10-07.

4.2.1.1.2.1.4 Testimony

[Coney 2004-09-22] Lillie Coney, testimony to EAC, available from http://vote.nist.gov/PanalistandPublicTestimony.html.

[Conrad 2004-09-22] Frederick Conrad, testimony to EAC, available from http://vote.nist.gov/PanalistandPublicTestimony.html.

[Deutsch 2004-09-21] Herb Deutsch, testimony to EAC, available from http://vote.nist.gov/PanalistandPublicTestimony.html.

[Fischer 2004-09-20] Eric A. Fischer, testimony to EAC, available from http://vote.nist.gov/PanalistandPublicTestimony.html.

[Gaston 2004-09-20] Charles A. Gaston, testimony to EAC, available from http://vote.nist.gov/PanalistandPublicTestimony.html.

[Golden 2004-09-22] Diane Cordry Golden, testimony to EAC, available from http://vote.nist.gov/PanalistandPublicTestimony.html.

[Jones 2004-09-20] Douglas W. Jones, testimony to EAC, available from http://www.cs.uiowa.edu/%7Ejones/voting/nist2004.shtml.

[Jones 2004-09-23] Douglas W. Jones, supplemental testimony to EAC, available from http://www.cs.uiowa.edu/%7Ejones/voting/nist2004supp.shtml.

[King 2004-09] Merle S. King, testimony to EAC, available from http://vote.nist.gov/PanalistandPublicTestimony.html.

[Noren 2004-09] Wendy S. Noren, testimony to EAC, available from http://vote.nist.gov/PanalistandPublicTestimony.html.

[Redish 2004-09-22] Janice Redish, testimony to EAC, available from http://vote.nist.gov/PanalistandPublicTestimony.html.

[Relton 2004-09-21] Joy Relton, testimony to EAC, available from http://vote.nist.gov/PanalistandPublicTestimony.html.

[Saltman 2004-09-20] Roy G. Saltman, testimony to EAC, available from http://vote.nist.gov/PanalistandPublicTestimony.html.

[Shamos 2004-09-20] Michael I. Shamos, testimony to EAC, available from http://vote.nist.gov/PanalistandPublicTestimony.html.

[Wallach 2004-09-20] Dan S. Wallach, testimony to EAC, available from http://vote.nist.gov/PanalistandPublicTestimony.html.

4.2.1.1.2.2 Standards architecture

NIST is recommending a reorganization of the VVSG to bring them in line with applicable standards practices that are abstracted from our years of association with ISO, W3C and other standards-creating organizations. This includes adding a section to define the meaning of conformance (called a conformance clause in ISO), identifying testable requirements as compliance points, and defining profiles, which allow requirements to vary as needed to accommodate variations in voting equipment.

Preferably, requirements should specify what (the desired performance), not how (a design to accomplish that). For example, a requirement that reads “single-bit errors shall be detected” is preferable to one that reads “products shall use memories with parity bits.” Profiles are created to resolve the conflict that occurs when the "what" depends on the "how". For example, the unstated assumption that the voting equipment would have an electronic memory at all requires placing the preceding example in a profile for electronic voting equipment.
Design-constraining requirements are controversial because vendors would like the freedom to provide the desired qualities / performance in different ways. However, in cases where vendors are unable to determine for themselves whether or not a given design is conforming, they may welcome design constraints as a way to avoid repeated failures and costly retesting of their products. Moreover, in cases where the desired quality is difficult to define abstractly, an enumeration of conforming cases may be the only practical alternative, particularly if there is only one design approach that is ever actually usable in practice. Some pragmatism will be required.

A vendor who is submitting a system for testing must provide an implementation statement that identifies exactly which profiles the system is asserted to support. Conformance tests may be catalogued according to which compliance points they exercise. The set of conformance tests appropriate to that claim may then be determined automatically. Upon passing those tests, the system may be qualified for only the claimed profiles.

Identified compliance points and a profiles mechanism in the VSS will facilitate traceability from state standards to the VSS. States will effectively define their own profiles over the VSS, adding compliance points they deem necessary without excessive repetition and revision of VSS text.

4.2.1.2 Testing

4.2.1.2.1 Purpose

The 2002 Voting Systems Standards define qualification testing as "the examination and testing of a computerized voting system by an Independent Test Authority (ITA) using qualification test standards to determine if the system complies with the qualification performance and test standards and with its own specifications. This process occurs prior to state certification."

The purpose of voting system (qualification) testing is to provide the states and other affected stakeholders with some level of assurance that a voting system is fit for use. States have the option to subject a voting system to additional scrutiny before purchasing and deploying it; however, most states require qualification by an ITA as an entry condition. *(Note: ITAs are also referred to as Voting System Testing Laboratories- VSTLs.)*

Even if procedural controls and audit trails ensured that any miscount would be detected, it could still be catastrophic for a state to have to rerun a compromised election and to remedy the faulty equipment. It is in the states' interests for the qualification process to eliminate voting systems that are not trustworthy before they are purchased and deployed.

4.2.1.2.2 Types of test methods

Traditionally, testing methods have been divided into black-box and white-box test design. Neither method has universal applicability; they are useful in the testing of different items.

Black-box testing is usually described as focusing on testing functional requirements, these requirements being defined in an explicit specification. It treats the item being tested as a "black box," with no examination being made of the internal structure or workings of the item. Rather, the nature of black-box testing is to develop and utilize detailed scenarios, or test cases. These test cases include specific sets of input to be applied to the item being tested. The output produced by the given input is then compared to a previously defined set of expected results.

White-box testing (sometimes called clear-box testing to suggest a more accurate metaphor) allows one to peek inside the "box," and focuses specifically on using knowledge of the internals of the item being tested to guide the testing procedure and the selection of test data. White-box

testing can discover extra non-specified functions for which black-box testing wouldn't know to look and can exercise data paths that would not have been exercised by a fixed test suite. Such extras can only be discovered by inspecting the internals.

Complimentary to any kind of testing is logic verification, in which formal methods are used to prove that the logic of the system satisfies certain assertions. When it is impractical to test every case in which a failure might occur, formal methods can be used to prove the correctness of the logic generally. However, verification is not a substitute for testing because there can be faults in a formal proof just as surely as there can be faults in a system. Used together, testing and verification can provide a high level of assurance that a system's logic is correct.

4.2.1.2.3 Repeatability and reproducibility

For qualification of voting systems to be consistent, fair, and meaningful, it is necessary to control variability in the conformity assessment system. Testing cannot be an afterthought to a standard: both the requirements to be tested and the methods by which they are to be tested must be specified with appropriate precision. The following hypothetical example illustrates the codependence of requirements and test methods.

Example text	Impact on testing
The unit shall respond to all user input in a timely fashion.	Vague requirement leaves tester in the position of determining what is considered "timely," creates opportunities for inconsistent evaluation and challenges by vendors.
The unit shall respond to all user input in 3 seconds or less.	Good requirement leading to pass-fail verdict. However, the test method to verify the requirement is undefined. Different testing authorities using different test methods may get different results. The vendor could challenge that the set of user inputs chosen by a VSTL is atypical of use in practice.
The VSTL shall measure and report the mean response time and worst response time over the following set of user inputs, employing the test ballot form defined in Section XYZ: opening the ballot; voting for one candidate in each contest; […]. Units with worst response time exceeding 3 seconds shall be disqualified.	In conjunction with the good requirement, this specified test method enables consistent, informative, and difficult-to-challenge results.

In Resolution #25-05, the TGDC requested that NIST perform a complete review and revision of requirements in the Voting Systems Standards to ensure that they are sufficiently precise to enable meaningful testing and to expand the testing standards to specify test methods for those requirements. This is a large undertaking and work on it continues. To date (2005-03-28), NIST has produced formal definitions for the terms that appear in vote data reports so that the accuracy of those terms in actual reports is well-defined; has defined abstract test cases pertaining to the

accumulation, counting, and reporting of votes; and has revised the text of relevant compliance points to improve their clarity and precision.

Additionally, in response to TGDC Resolution #27-05, NIST recommends eliminating the provision in the 2002 VSS for qualification of voting systems that do not conform to the requirements.[2] One member of the TGDC indicated that this provision was of historical origin and is of no further use.

4.2.1.3 Transparency

The public must also be assured that the voting system is fit for use. This can occur vicariously, through trust in the VSTL and election officials; indirectly, through verification that the qualification process was responsibly executed; directly, through election verification; or through a combination of these.

In Resolution #28-05, the TGDC requested that NIST recommend standards on data to be provided, called a "Public Information Package," that must be publicly available and published as evidence that the qualification process was responsibly executed. These requirements now appear in the draft Testing document (Vol. III) and will continue to be expanded as the testing standards are expanded.

With respect to election verification, the Security and Transparency Subcommittee is currently drafting recommendations pertaining to Directly Verifiable (DV) systems and Indirectly Verifiable (IV) systems.

4.2.1.4 Coding conventions and code reviews

Volume 1, Section 4.2 and Volume 2, Section 5.4 of the 2002 Voting Systems Standards define coding conventions and a source code review to be conducted by ITAs. Vendors are permitted to use current best practices in lieu of the coding conventions defined in the VSS; however, the coding conventions in the VSS are out of date, and if followed, could do more harm than good.

The coding conventions are a means to the end of facilitating ITA evaluation of the code's correctness to some level of assurance beyond that provided by black-box testing. That evaluation is underspecified in the 2002 VSS, yielding a cart-before-horse situation in which adherence to the coding conventions could be verified much more rigorously than the correctness of the software.

In Resolution #29-05, the TGDC requested that NIST:

– Recommend standards to be used in evaluating the correctness of voting system logic, including but not limited to software implementations, and

[2] Volume 2, Appendix B, Section B.5: "Any uncorrected deficiency that does not involve the loss or corruption of voting data shall not necessarily be cause for rejection."

– Evaluate the 2002 VSS software coding standards with respect to their applicability to the recommended standards, and either revise them, delete them, or recommend new software coding standards, as appropriate.

To date (2005-03-25), the former is included in the draft Testing document (Vol. III), while the latter is provided as change-tracked text for Vol. I Ch. 4 and Vol. II Ch. 5 of the 2002 VSS.

Coding conventions addressing the need for integrity in voting software have been retained, expanded, and made mandatory, while stylistic conventions that are made redundant by more recent, publicly available coding conventions have been removed in favor of the published conventions. Whether the coding conventions addressing integrity can also be replaced by recent, publicly available coding conventions for high-integrity software is yet to be determined.

One possibly controversial recommendation included in the changes to the software standards is to require the use of a programming language that supports structured exception handling. This rules out the C language, which remains in wide use, and forces a migration to a descendant language, namely C++, C#[3] or Java. Similarly, older versions of Visual Basic that lacked structured exception handling are superseded by Visual Basic .NET.

This recommendation is induced by existing requirements in the VSS, namely:

– I.2.2.5.2.2.g: Nested error conditions shall be corrected in a controlled sequence such that system status shall be restored to the initial state existing before the first error occurred.
– I.4.2.3.e: Each module shall have a single entry point, and a single exit point, for normal process flow. The exception for the exit point is where a problem is so severe that execution cannot be resumed. In this case, the design must explicitly protect all recorded votes and audit log information and must implement formal exception handlers provided by the language.

It appears to be the intent of these requirements that the voting system software should (A) exhibit behaviors that are representative of structured exception handling, and (B) accomplish these using "formal exception handlers provided by the language." In context, this is puzzling, since the VSS specifically allowed languages that did not support any semblance of formal exceptions. However, as of 2005, programming languages supporting structured exceptions are widely available and widely used, and they contain other refinements and evolutionary advances, relative to their exceptionless ancestors, that contribute to enhanced software integrity, maintainability, and understandability. To require their use now is in the same spirit of best practices as the VSS' 1990 requirement for structured control constructs, which has since then been rendered redundant by the virtual extinction of programming languages that do not include those constructs (and of programmers who fail to use them).

Though potentially painful, the migration from languages not supporting structured exceptions is facilitated by closely related languages that evolved from one another: C and C++, C# or Java,

[3] Commercial equipment and materials are identified in order to describe certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

Visual Basic and Visual Basic .NET. Nevertheless, if the requirement for structured exception handling should be removed to avoid forcing such migrations, it would not be fatal to the guidelines.

4.2.1.5 Quality assurance and configuration management

Volume 1, Sections 7 and 8 and Volume 2, Section 7 of the 2002 VSS require the vendor to follow certain quality assurance and configuration management practices and require the ITA to conduct several audits and documentation reviews to ensure that they were followed. The quality assurance and configuration management requirements in the VSS are a means to the end of ensuring that the vendor has followed responsible engineering practices in general, and are not necessarily the best or most up-to-date guidelines for that purpose.

In Resolution #30-05, the TGDC requested that NIST review and analyze quality assurance and configuration management standards and recommend changes to the VVSG based on that analysis.

Since the Voting Systems Standards were first issued, it has become possible for vendors to be certified under ISO 9000 and/or appraised under CMMI.[4] It is not clear whether a separate standard for voting system vendors, in lieu of requiring ISO 9000 certification to a scope of operations appropriate to the purpose of developing voting systems, is any longer necessary or desirable. However, at its January 2005 meeting, the TGDC expressed fear over the expense and administrative burden involved in ISO 9000 compliance. NIST has not yet completed the review and analysis of related standards to determine whether a less expensive alternative exists or whether the existing standards could be retained without sacrificing quality.

4.2.2 Product Standard

The following requirements have been extracted from the 2002 VSS and (in one case) IEEE DRAFT P1583/D5.3.2b, 2005-01-04.

Most requirements have been refactored from the structured text of the 2002 VSS to make them self-contained compliance points. Some have undergone additional rewording to improve their precision and clarify them. This work is still ongoing.

The compliance points are organized according to the process model rev. 2005-02-23. *Thus, each compliance point should be read as if it has a Process: field referring to the activity or activities in the process model indicated by the subsection title.*

Except where otherwise specified, all compliance points in this document implicitly have Responsible Entity: Voting System Vendors

[4] Capability Maturity Model Integration, http://www.sei.cmu.edu/cmmi/.

Requirements for Test Labs have been moved to 4.2.3 (CRT Testing Pieces) and there are only two requirements applicable to Voting Officials.

4.2.2.1 General requirements

4.2.2.1.1 All systems shall support the gathering of votes using all voting variations indicated in the Implementation Statement. {Extrapolated from VSS I.2.2.8.2 and I.2.4}

4.2.2.1.2 All systems shall achieve an error rate of no more than one in 10,000,000 ballot positions. {Extrapolated from VSS I.3.2.1}

4.2.2.1.3 All systems shall be capable of generating the required reports. {Generalized from many VSS requirements}

4.2.2.1.4 All systems shall be auditable by election officials. {Generalized from many VSS requirements}

4.2.2.1.5 All systems shall maintain the integrity of voting and audit data, including Cast Vote Records, during an election and for a period of at least 22 months afterward. {Reworded from VSS I.2.11}Responsible Entity: Voting System Vendors, Voting Officials

4.2.2.1.6 All systems shall maximize interoperability and integratability with other systems and/or components of other systems. {Generalized from Steve Freeman interpretation of database design requirements in VSS I.2.2.6, TGDC Resolution #23-05, and some state RFP(s)}

4.2.2.2 Prepare for election

[Discussion: There are significant variations among the election laws of the 50 states with respect to permissible ballot contents, voting options, and the associated ballot counting logic. {VSS 2.2.8.2}]

4.2.2.2.1 The Election Management System shall support all voting variations indicated in the Implementation Statement. {Extrapolated from VSS I.2.2.8.2}

4.2.2.3 Define precincts

4.2.2.3.1 In systems claiming conformance to the *Split precincts* profile, the Election Management System shall support split precincts. {Extrapolated from VSS I.2.2.8.2}

4.2.2.4 Program election

[Discussion: 2002 VSS I.4.4.1 defines pre-election audit records. These should be detailed under Principle 2.8. What is within scope of Principle2. 4 is the production of reports from those records. This requirement appears under 4.2.2.16 below.]

4.2.2.4.1 In systems claiming conformance to the *Closed primaries* profile, the Election Management System shall support closed primaries, partisan and non-partisan offices. {Extrapolated from VSS I.2.2.8.2}

4.2.2.4.2 In systems claiming conformance to the *Open primaries* profile, the Election Management System shall support open primaries, partisan and non-partisan offices. {Extrapolated from VSS I.2.2.8.2}

4.2.2.4.3 In systems claiming conformance to the *Write-ins* profile, the Election Management System shall support write-in voting. {Extrapolated from VSS I.2.2.8.2}

4.2.2.4.4 In systems claiming conformance to the *Ballot rotation* profile, the Election Management System shall support ballot rotation. {Extrapolated from VSS I.2.2.8.2}

4.2.2.4.5 In systems claiming conformance to the *Straight party voting* profile, the Election Management System shall support straight party voting. {Extrapolated from VSS I.2.2.8.2}

4.2.2.4.6 In systems claiming conformance to the *Cross-party endorsement* profile, the Election Management System shall support cross-party endorsement. {Extrapolated from VSS I.2.2.8.2}

4.2.2.4.7 In systems claiming conformance to the *Cross-party endorsement* profile, the Election Management System shall support the endorsement of a given candidate by two or three different political parties, and may support more. {Clarification or extension of existing requirement}

4.2.2.4.8 In systems claiming conformance to the *N of M voting* profile, the Election Management System shall support N of M voting. {Extrapolated from VSS I.2.2.8.2}

4.2.2.4.9 In systems claiming conformance to the *Cumulative voting* profile, the Election Management System shall support cumulative voting. {Extrapolated from VSS I.2.2.8.2}

4.2.2.4.10 In systems claiming conformance to the *Ranked order voting* profile, the Election Management System shall support ranked order voting. {Extrapolated from VSS I.2.2.8.2}

4.2.2.4.11 In systems claiming conformance to the *Provisional / challenged ballots* profile, the Election Management System shall support provisional/challenged ballots. {Extrapolated from VSS I.2.2.8.2}

4.2.2.4.12 In systems claiming conformance to the *Review-required ballots* profile, the Election Management System shall support review-required ballots. {Extrapolated from VSS I.2.5.2}

4.2.2.4.13 The interoperability requirement for election programming data may be met by providing the capability to export election programming data in a non-proprietary, open standard format. {Drill-down from TGDC Resolution #23-05}

4.2.2.4.14 The interoperability requirement for election programming data may be met by storing election programming data in a documented schema in a COTS or non-proprietary, open source database in such a manner that other applications can read and interpret the data. {Drill-down from VSS I.2.2.6 and Steve Freeman interpretation}

4.2.2.5 Prepare for voting (precinct)

4.2.2.5.1 Test precinct equipment (precinct)

[Discussion: 2002 VSS I.2.3.5 and I.4.4.2 define system readiness tests and audit records (precinct). These should be detailed under Principle 2.8. What is within scope of Principle 2.4 is the production of reports from those records. This requirement appears under 4.2.2.16 *below.]*

4.2.2.5.2 Open poll

[Discussion: 2002 VSS I.2.4.1 defines tests to be performed when polls are opened. These should be detailed under Principle2.8.]

4.2.2.5.2.1 All systems shall support opening the polls. {VSS I.2.4}

4.2.2.6 Prepare for voting (central)

4.2.2.6.1 Test central equipment (central)

[Discussion: 2002 VSS I.2.3.6 and I.4.4.2 define system readiness tests and audit records (central). These should be detailed under Principle 2.8. What is within scope of Principle 2.4 is the production of reports from those records. This requirement appears under 4.2.2.16 below.]

4.2.2.7 Gather in-person vote

[Discussion: 2002 VSS I.4.4.3 defines in-process audit records. These should be detailed under Principle 8. What is within scope of Principle 2.4 is the production of reports from those records. This requirement appears under 4.2.2.16 below.]

4.2.2.7.1 All systems shall support casting a ballot. {VSS I.2.4}

4.2.2.7.2 Systems claiming conformance to the *DRE* profile shall support activating the ballot. {VSS I.2.4}

4.2.2.7.3 To activate the ballot, systems claiming conformance to the *DRE* profile shall enable election officials to control the content of the ballot presented to the voter, whether presented in printed form or electronic display, such that each voter is permitted to record votes only in contests in which that voter is authorized to vote. {VSS I.2.4.2.a}

4.2.2.7.4 To activate the ballot, systems claiming conformance to the *DRE* profile shall allow each eligible voter to cast a ballot. {VSS I.2.4.2.b}

[Discussion: This FR and subsequent overlap with Principle 2; put here for now.]

4.2.2.7.5 To activate the ballot, systems claiming conformance to the *DRE* profile shall prevent a voter from voting on a ballot to which he or she is not entitled. {VSS I.2.4.2.c}

4.2.2.7.6 To activate the ballot, systems claiming conformance to the *DRE* profile shall prevent a voter from casting more than one ballot in the same election. {VSS I.2.4.2.d}

[Discussion: Deleted 2.4.2.e: redundant.]

4.2.2.7.7 To activate the ballot, systems claiming conformance to the *DRE* and *Open primaries* or *Closed primaries* profiles shall enable the selection of the ballot that is appropriate to the party affiliation declared by the voter in a primary election. {VSS I.2.4.2.f}

4.2.2.7.8 To activate the ballot, systems claiming conformance to the *DRE* profile shall activate all portions of the ballot upon which the voter is entitled to vote. {VSS I.2.4.2.g}

4.2.2.7.9 To activate the ballot, systems claiming conformance to the *DRE* profile shall disable all portions of the ballot upon which the voter is not entitled to vote. {VSS I.2.4.2.h}

4.2.2.7.10 To facilitate casting a ballot, all systems shall record the selection and non-selection of individual vote choices for each contest and ballot measure. {VSS I.2.4.3.1.c}

4.2.2.7.11 To facilitate casting a ballot, systems claiming conformance to the *Write-ins* profile shall record the voter's selection of candidates whose names do not appear on the ballot, if permitted under State law, and record as many write-in votes as the number of candidates the voter is allowed to select, per the definition of N(*r*) in Vol. III, Logic Verification. {VSS I.2.4.3.1.d}

4.2.2.7.12 Systems claiming conformance to the *DRE* profile shall allow the voter to select his or her preferences on the ballot in any legal number and combination. {VSS I.2.4.3.3.c}

[Discussion: Lots of other reqs in I.2.4.3.3 are usability.]

4.2.2.7.13 Systems claiming conformance to the *DRE* profile shall prevent the voter from overvoting. {VSS I.2.4.3.3.f}

4.2.2.7.14 Systems claiming conformance to the *DRE* profile shall prevent modification of the voter's vote after the ballot is cast. {VSS I.2.4.3.3.n}

4.2.2.7.15 Systems claiming conformance to the *DRE* profile shall verify (i.e., actively check and confirm) the correct addition of voter selections to the memory components of the device. {VSS I.3.2.4.3.3.c}

4.2.2.7.16 Systems claiming conformance to the *Optical Scan* profile shall allow the voter to mark the ballot to register a vote. {VSS I.2.4.3.2.1.b}

4.2.2.7.17 Systems claiming conformance to the *Punchcard* profile shall allow the voter to punch the ballot to register a vote. {VSS I.2.4.3.2.1b}

4.2.2.7.18 In systems claiming conformance to the *DRE* profile, the vote-gathering functionality of each DRE shall support all voting variations indicated in the Implementation Statement. {Extrapolated from VSS I.2.2.8.2}

4.2.2.7.19 In systems claiming conformance to the *Closed primaries* and *DRE* profiles, the ballot presentation, voting, and recording functionality of each DRE shall support closed primaries, partisan and non-partisan offices. {Extrapolated from VSS I.2.2.8.2}

4.2.2.7.20 In systems claiming conformance to the *Open primaries* and *DRE* profiles, the ballot presentation, voting, and recording functionality of each DRE shall support open primaries, partisan and non-partisan offices. {Extrapolated from VSS I.2.2.8.2}

4.2.2.7.21 In an open primary on a DRE system, the voter shall be allowed to choose a party affiliation at the time of voting and vote the appropriate ballot form in privacy (i.e., the choice of affiliation shall be private as well as the ballot). {Clarification or extension of existing requirements}

[Discussion: FIXME: This belongs in privacy section.]

4.2.2.7.22 In systems claiming conformance to the *Write-ins* and *DRE* profiles, the ballot presentation, voting, and recording functionality of each DRE shall support write-in voting. {Extrapolated from VSS I.2.2.8.2}

4.2.2.7.23 In systems claiming conformance to the *Ballot rotation* and *DRE* profiles, the ballot presentation, voting, and recording functionality of each DRE shall support ballot rotation. {Extrapolated from VSS I.2.2.8.2}

4.2.2.7.24 DRE systems that enable ballot rotation in a given contest shall alter the ordering of candidates or choices in such a manner that no candidate or choice shall ever have appeared in any particular ballot position two or more times more often than any other. {Clarification or extension of existing requirements}

4.2.2.7.25 In systems claiming conformance to the *Straight party voting* and *DRE* profiles, the ballot presentation, voting, and recording functionality of each DRE shall support straight party voting. {Extrapolated from VSS I.2.2.8.2}

4.2.2.7.26 In systems claiming conformance to the *Cross-party endorsement* and *DRE* profiles, the ballot presentation, voting, and recording functionality of each DRE shall support cross-party endorsement. {Extrapolated from VSS I.2.2.8.2}

4.2.2.7.27 In systems claiming conformance to the *Cross-party endorsement* and *DRE* profiles, the ballot presentation, voting, and recording functionality of each DRE shall support the endorsement of a given candidate by two or three different political parties, and may support more. {Clarification or extension of existing requirement}

4.2.2.7.27 In systems claiming conformance to the *Split precincts* and *DRE* profiles, the ballot presentation, voting, and recording functionality of each DRE shall support split precincts. {Extrapolated from VSS I.2.2.8.2}

4.2.2.7.28 In systems claiming conformance to the *N of M voting* and *DRE* profiles, the ballot presentation, voting, and recording functionality of each DRE shall support N of M voting. {Extrapolated from VSS I.2.2.8.2}

4.2.2.7.29 In systems claiming conformance to the *Cumulative voting* and *DRE* profiles, the ballot presentation, voting, and recording functionality of each DRE shall support cumulative voting. {Extrapolated from VSS I.2.2.8.2}

4.2.2.7.30 In systems claiming conformance to the *Ranked order voting* and *DRE* profiles, the ballot presentation, voting, and recording functionality of each DRE shall support ranked order voting. {Extrapolated from VSS I.2.2.8.2}

4.2.2.7.31 In systems claiming conformance to the *Provisional / challenged ballots* and *DRE* profiles, the ballot presentation, voting, and recording functionality of each DRE shall support provisional/challenged ballots. {Extrapolated from VSS I.2.2.8.2}

4.2.2.7.32 In systems claiming conformance to the *Review-required ballots* and *DRE* profiles, the ballot presentation, voting, and recording functionality of each DRE shall support review-required ballots. {Extrapolated from VSS I.2.5.2}

4.2.2.8 Accept ballot

4.2.2.8.1 Systems claiming conformance to the *Precinct count* and either the *Optical Scan* or *Punchcard* profile shall allow either the voter or the appropriate election official to place the voted ballot into the ballot counting device. {VSS I.2.4.3.2.1.c}

4.2.2.8.2 Systems claiming conformance to the *Central count* and either the *Optical Scan* or *Punchcard* profile shall allow either the voter or the appropriate election official to place the voted ballot into a secure receptacle. {VSS I.2.4.3.2.1.c}

4.2.2.8.3 For systems claiming conformance to the *DRE* profile, the acceptable voting system error rate (no more than one in 10,000,000 ballot positions) applies to recording the voter selections of candidates and contests into voting data storage. (From VSS I.3.2.1.b.1)

4.2.2.8.4 For systems claiming conformance to the *DRE* profile, the acceptable voting system error rate (no more than one in 10,000,000 ballot positions) applies to recording voter selections of candidates and contests into ballot image storage independently of voting data storage. (From VSS I.3.2.1.b.2)

[Discussion: This relates to the "separate path" design requirement below, to be revised.]

[Discussion: The following design requirements should be obsoleted by recommendations from STS. Until then, it is important to retain some requirements for meaningful auditability, and these are the best we have at the moment.]

4.2.2.8.5 Systems claiming conformance to the *DRE* profile shall maintain Cast Vote Records using a process and storage location that differs from the main vote detection, interpretation, processing, and reporting path. {Reworded from VSS I.2.2.4.2}

4.2.2.8.6 Systems claiming conformance to the *DRE* profile shall provide at least two processes that record the voter's selections that, to the extent possible, are isolated from each other. {VSS I.3.2.4.3.2.c.1}

4.2.2.8.7 Systems claiming conformance to the *DRE* profile shall record and retain redundant copies of the original ballot image. {VSS I.2.2.2.2}

4.2.2.9 Count (precinct count) + Count (central)

[Discussion: The following requirements apply equally to counting that occurs in the precinct and in the central location.]

4.2.2.9.1 All tabulators shall support all voting variations indicated in the Implementation Statement. {VSS I.2.2.8.1 plus I.2.2.8.2}

4.2.2.9.2 In systems claiming conformance to the *Closed primaries* profile, the vote tabulating functionality of each voting device, vote count server, or other devices shall support closed primaries, partisan and non-partisan offices. {VSS I.2.2.8.1 plus I.2.2.8.2}

4.2.2.9.3 In systems claiming conformance to the *Open primaries* profile, the vote tabulating functionality of each voting device, vote count server, or other devices shall support open primaries, partisan and non-partisan offices. {VSS I.2.2.8.1 plus I.2.2.8.2}

4.2.2.9.4 In systems claiming conformance to the *Write-ins* profile, the vote tabulating functionality of each voting device, vote count server, or other devices shall support write-in voting. {VSS I.2.2.8.1 plus I.2.2.8.2}

4.2.2.9.5 In systems claiming conformance to the *Ballot rotation* profile, the vote tabulating functionality of each voting device, vote count server, or other devices shall support ballot rotation. {VSS I.2.2.8.1 plus I.2.2.8.2}

4.2.2.9.6 In systems claiming conformance to the *Straight party voting* profile, the vote tabulating functionality of each voting device, vote count server, or other devices shall support straight party voting. {VSS I.2.2.8.1 plus I.2.2.8.2}

4.2.2.9.7 A straight party vote shall be counted as a vote in favor of all candidates endorsed by the chosen party in each contest in which the voter does not cast an explicit vote. {Clarification or extension of existing requirements}

4.2.2.9.8 An explicit vote in a given contest takes precedence over a straight party vote and nullifies the effect of a straight party vote for only that contest. {Clarification or extension of existing requirements}

4.2.2.9.9 In systems claiming conformance to the *Cross-party endorsement* profile, the vote tabulating functionality of each voting device, vote count server, or other devices shall support cross-party endorsement. {VSS I.2.2.8.1 plus I.2.2.8.2}

4.2.2.9.10 In systems claiming conformance to the *Cross-party endorsement* profile, the vote tabulating functionality of each voting device, vote count server, or other devices shall support the counting of straight party votes when a given candidate is

endorsed by two or three different political parties, and may support more. {Clarification or extension of existing requirement}

4.2.2.9.11 In systems claiming conformance to the *Split precincts* profile, the vote tabulating functionality of each voting device, vote count server, or other devices shall support split precincts. {VSS I.2.2.8.1 plus I.2.2.8.2}

4.2.2.9.12 In systems claiming conformance to the *N of M voting* profile, the vote tabulating functionality of each voting device, vote count server, or other devices shall support N of M voting. {VSS I.2.2.8.1 plus I.2.2.8.2}

4.2.2.9.13 In systems claiming conformance to the *Cumulative voting* profile, the vote tabulating functionality of each voting device, vote count server, or other devices shall support cumulative voting. {VSS I.2.2.8.1 plus I.2.2.8.2}

4.2.2.9.14 In systems claiming conformance to the *Ranked order voting* profile, the vote tabulating functionality of each voting device, vote count server, or other devices shall support ranked order voting. {VSS I.2.2.8.1 plus I.2.2.8.2}

4.2.2.9.15 In systems claiming conformance to the *Provisional / challenged ballots* profile, the vote tabulating functionality of each voting device, vote count server, or other devices shall support provisional/challenged ballots. {VSS I.2.2.8.1 plus I.2.2.8.2}

4.2.2.9.16 In systems claiming conformance to the *Review-required ballots* profile, the vote tabulating functionality of each voting device, vote count server, or other devices shall support review-required ballots. {Extrapolated from VSS I.2.5.2}

4.2.2.9.17 For systems claiming conformance to the *Optical Scan* or *Punchcard* profile, the acceptable voting system error rate (no more than one in 10,000,000 ballot positions) applies to scanning ballot positions on paper ballots to detect selections for individual candidates and contests. (From VSS I.3.2.1.a.1)

[Discussion: A public comment has been received that recommends adjusting the following two requirements to quantify permissible deviations from the target marking area instead of merely conforming to vendor specifications. This comment is available at http://vote.nist.gov/ecposstatements/AVANTEACCURACY.doc. NIST has insufficient experience with the relevant hardware to determine whether the suggested specifications are reasonable to implement and test without additional research. The TGDC is requested to examine this issue and alter the following requirements as needed.]

4.2.2.9.18 For systems claiming conformance to the *Optical Scan* or *Punchcard* profile, the acceptable voting system error rate (no more than one in 10,000,000 ballot positions) applies to the detection of punches or marks that conform to vendor specifications. {VSS I.3.2.5.2.a and I.3.2.6.1.1}

[*Discussion: Vendor specifications may not reflect the behavior of actual voters. Quantify the required performance. (Requires research with human subjects)*]

4.2.2.9.19 Systems claiming conformance to the *Optical Scan* or *Punchcard* profile shall ignore, and not record, extraneous perforations, smudges, and folds. {VSS I.3.2.5.2.b}

[Discussion: Quantify "extraneous" – how big does an extraneous smudge get before it's considered an intentional mark? (Requires research with human subjects – need to know if the marks were intentional)]

4.2.2.9.20 For systems claiming conformance to the *Optical Scan* or *Punchcard* profile, the acceptable voting system error rate (no more than one in 10,000,000 ballot positions) applies to conversion of selections detected on paper ballots into digital data. (From VSS I.3.2.1.a.2 and I.3.2.6.1.1)

4.2.2.10 Count (precinct count)

4.2.2.10.1 For systems claiming conformance to the *Precinct count* profile, the acceptable voting system error rate (no more than one in 10,000,000 ballot positions) applies to consolidation of vote selection data from multiple precinct-based systems to generate jurisdiction-wide vote counts, including storage and reporting of the consolidated vote data. {Reworded from VSS I.3.2.1}

4.2.2.11 Wrap up voting (precinct)

4.2.2.11.1 Close polls

4.2.2.11.1.1 Systems claiming conformance to the *Precinct count* profile shall provide designated functions for generating post-election reports. {Reworded from VSS I.2.5}

4.2.2.11.1.2 Systems claiming conformance to the *Precinct count* profile shall consolidate the data contained in each unit into a single report for the polling place when more than one voting machine or precinct tabulator is used. {Reworded from VSS I.2.5.3.2}

4.2.2.11.1.3 Systems claiming conformance to the *DRE* profile shall, if the consolidation of polling place data is done locally, perform this consolidation in a time not to exceed 5 minutes for each device in the polling place. {Reworded from VSS I.3.2.6.2.1}

[Discussion: For requirements on report content see 4.2.2.16.]

4.2.2.11.1.4 Systems claiming conformance to the *Precinct count* profile shall provide designated functions for closing the polling place. {Reworded from VSS I.2.5}

4.2.2.11.1.5 Systems claiming conformance to the *Precinct count* profile shall provide a means to prevent the further casting of ballots once the polling place has closed. {Reworded from VSS I.2.5.1.a)

4.2.2.11.1.6 Systems claiming conformance to the *Precinct count* profile shall provide an internal test that verifies that the prescribed closing procedure has been followed and that the device status is normal. {Reworded from VSS I.2.5.1.b)

4.2.2.11.1.7 Systems claiming conformance to the *Precinct count* profile shall include a visible indication of system status (i.e., whether the polls are opened or closed). {Reworded from VSS I.2.5.1.c)

4.2.2.11.1.8 Systems claiming conformance to the *Precinct count* profile shall provide a means to produce a diagnostic test record that verifies the sequence of events and indicates that the extraction of voting data has been activated. {Reworded from VSS I.2.5.1.d)

4.2.2.11.1.9 Systems claiming conformance to the *Precinct count* profile shall provide a means to preclude the unauthorized reopening of the polls once the poll closing has been completed for that election. {Reworded from VSS I.2.5.1.e)

4.2.2.12 Diagnose and correct problem (precinct) + Diagnose and correct problem (central)

4.2.2.12.1 Any discrepancy in reports, regardless of source, shall be resolvable to a procedural error, to the failure of a non-memory device, or to an external cause. {Reworded from VSS I.3.2.6.2.2}

[Discussion: Important requirement, but not testable: if you are here, your reports are inconsistent, so you fail some other test.]

4.2.2.13 Deliver / transmit ballots, ballot images and/or precinct totals to central

4.2.2.13.1 All systems shall ensure that extracted or duplicated information, including Cast Vote Records extracted from DRE machines, is identical to that on the original storage medium. {Reworded from Section 5.6.9.2, Paragraph k of IEEE DRAFT P1583/D5.3.2b, 2005-01-04.}[5]

4.2.2.13.2 All electronic systems shall verify (i.e., actively check and confirm) that information as extracted or duplicated to machine-readable media is identical to that on the original storage medium.

[5] This material is from an unapproved draft of a proposed IEEE Standard, P1583. As such, the material is subject to change in the final standard. Because this material is from an unapproved draft, the IEEE recommends that it not be utilized for any conformance/compliance purposes. It is used at your own risk.

4.2.2.13.3 All systems shall prevent data from being altered or destroyed by the transmission of results over telecommunications lines, including data in transportable memory. {Reworded from VSS I.2.5.3.1 and I2.5.3.2d}

4.2.2.13.4 Transmitting results over telecommunications lines shall not result in modifications to any vote data in the sending system.

4.2.2.14.4 All electronic systems shall ensure that information received by transmission over telecommunications lines is identical to what was sent.

4.2.2.14.5 The acceptable voting system error rate (no more than one in 10,000,000 ballot positions) applies to the transmission of data over telecommunications lines. {VSS I.5.2.1}

4.2.2.15 Count (central)

4.2.2.15.1 For systems claiming conformance to the *Central count* profile, the acceptable voting system error rate (no more than one in 10,000,000 ballot positions) applies to consolidation of vote selection data from multiple counting devices to generate jurisdiction-wide vote counts, including storage and reporting of the consolidated vote data. {Reworded from VSS I.3.2.1}

4.2.2.16 Report

4.2.2.16.1 All systems shall produce reports that account for all votes on all accepted ballots.

4.2.2.16.2 These reports shall be completely consistent and error-free, with no discrepancy among reports of voting device data at any level. {Reworded from VSS I.3.2.6.2.2, extended to all systems}

[Discussion: The following compliance points were distilled and refactored from overlapping, subtly differing requirements appearing several places in Chapters 2 and 4 of the 2002 VSS, including: I.2.2.2.1.c (produce an accurate report of all votes cast), I.2.2.6.h (printed report of everything in I.2.5), I.2.2.9 (ballot counter), I.2.5.2 (means to consolidate vote data), I.2.5.3.1.a (geographic reporting), I.2.5.3.1.b (printed report of number of ballots counted by each tabulator), I.2.5.3.1.c (contest results, overvotes, and undervotes for each tabulator), I.2.5.3.1.d (consolidated reports including other data sources), I.4.4.4.a (number of ballots cast, using each ballot configuration, by tabulator, precinct, and political subdivision), I.4.4.4.b (candidate and measure totals for each contest, by tabulator), I.4.4.4.c (number of ballots read within each precinct and for additional jurisdictional levels, by configuration, including separate totals for each party in primary elections), I.4.4.4.d (separate accumulation of overvotes and undervotes for each contest, by tabulator, precinct, and additional jurisdictional levels), and I.4.4.4.e (for paper-based systems, the total number of ballots both processed and unprocessable, and the total number of cards read).]

[Discussion: The following requirements depend on specific, new definitions of cast ballot, read ballot, and counted ballot that appear in the glossary.]

4.2.2.16.3 Systems claiming conformance to the *Optical Scan* or *Punchcard* profile shall report the total number of ballots *cast* at the precinct, election district, and jurisdiction reporting levels, by configuration.

4.2.2.16.4 Systems claiming conformance to the *Optical Scan* or *Punchcard* profile and the *Provisional / challenged ballots* profile shall report the total number of provisional ballots *cast* at the precinct, election district, and jurisdiction reporting levels, by configuration.

4.2.2.16.5 All systems shall report the total number of ballots *read* at each reporting level (tabulator, precinct, election district, and jurisdiction), by configuration.

4.2.2.16.6 Systems claiming conformance to the *Optical Scan* or *Punchcard* profile shall, if there are multiple card ballots, report the total number of cards read at the precinct, election district, and jurisdiction reporting levels, by configuration.

4.2.2.16.7 Systems claiming conformance to the *Closed primaries* or *Open primaries* profiles shall include separate totals for each party in primary elections.

4.2.2.16.8 Systems claiming conformance to the *Provisional / challenged ballots* profile shall report the total number of provisional ballots *read* at each reporting level (tabulator, precinct, election district, and jurisdiction), by configuration.

4.2.2.16.9 All systems shall report the total number of ballots *counted* at each reporting level (tabulator, precinct, election district, and jurisdiction), by configuration.

4.2.2.16.10 Systems claiming conformance to the *Closed primaries* or *Open primaries* profiles shall include separate totals for each party in primary elections.

4.2.2.16.11 Systems claiming conformance to the *Provisional / challenged ballots* profile shall report the total number of provisional ballots *counted* at each reporting level (tabulator, precinct, election district, and jurisdiction), by configuration.

4.2.2.16.12 All systems shall report the candidate and measure vote totals for each N-of-M or cumulative voting contest, at each reporting level (tabulator, precinct, election district, and jurisdiction), per the definition of $T(c,j,r,t_E)$ in Vol. III, Logic Verification.

[Discussion: N-of-M includes the most common type of contest, 1-of-M.]

4.2.2.16.13 Systems claiming conformance to the *In-person voting* profile shall include in-person votes in the consolidated report of vote totals.

4.2.2.16.14 Systems claiming conformance to the *Absentee voting* profile shall include absentee votes in the consolidated report of vote totals.

4.2.2.16.15 Systems claiming conformance to the *Provisional / challenged ballots* profile shall include votes from accepted provisional / challenged ballots in the consolidated report of vote totals.

4.2.2.16.16 Systems claiming conformance to the *Review-required ballots* profile shall include votes from accepted reviewed ballots in the consolidated report of vote totals.

4.2.2.16.17 All systems shall report the number of counted ballots for each N-of-M or cumulative voting contest, at each reporting level (tabulator, precinct, election district, and jurisdiction), per the definition of $K(j,r,t_E)$ in Vol. III, Logic Verification.

4.2.2.16.18 Systems claiming conformance to the *In-person voting* profile shall include in-person votes in the consolidated report of counted ballots.

4.2.2.16.19 Systems claiming conformance to the *Absentee voting* profile shall include absentee votes in the consolidated report of counted ballots.

4.2.2.16.20 Systems claiming conformance to the *Provisional / challenged ballots* profile shall include votes from accepted provisional / challenged ballots in the consolidated report of counted ballots.

4.2.2.16.20 Systems claiming conformance to the *Review-required ballots* profile shall include votes from accepted reviewed ballots in the consolidated report of counted ballots.

4.2.2.16.21 All systems shall report the number of overvotes for each N-of-M or cumulative voting contest, at each reporting level (tabulator, precinct, election district, and jurisdiction), per the definition of $O(j,r,t_E)$ in Vol. III, Logic Verification.

4.2.2.16.22 Systems claiming conformance to the *In-person voting* profile shall include in-person votes in the consolidated report of overvotes.

4.2.2.16.23 Systems claiming conformance to the *Absentee voting* profile shall include absentee votes in the consolidated report of overvotes.

4.2.2.16.24 Systems claiming conformance to the *Provisional / challenged ballots* profile shall include votes from accepted provisional / challenged ballots in the consolidated report of overvotes.

4.2.2.16.25 Systems claiming conformance to the *Review-required ballots* profile shall include votes from accepted reviewed ballots in the consolidated report of overvotes.

4.2.2.16.26 All systems shall report the number of undervotes for each N-of-M or cumulative voting contest, at each reporting level (tabulator, precinct, election district, and jurisdiction), per the definition of $U(j,r,t_E)$ in Vol. III, Logic Verification.

4.2.2.16.27 Systems claiming conformance to the *In-person voting* profile shall include in-person votes in the consolidated report of undervotes.

4.2.2.16.28 Systems claiming conformance to the *Absentee voting* profile shall include absentee votes in the consolidated report of undervotes.

4.2.2.16.29 Systems claiming conformance to the *Provisional / challenged ballots* profile shall include votes from accepted provisional / challenged ballots in the consolidated report of undervotes.

4.2.2.16.30 Systems claiming conformance to the *Review-required ballots* profile shall include votes from accepted reviewed ballots in the consolidated report of undervotes.

4.2.2.16.30 Systems claiming conformance to the *Ranked order voting* profile shall report the candidate and measure vote totals for each ranked order contest for each round of voting/counting at the jurisdiction level.

[Discussion: This requirement is minimal. Since ranked order voting is not currently in wide use, it is not clear whether a count must be reported for each permutation of choices, how bogus orderings are reported, or how it would be done at multiple reporting levels.]

4.2.2.16.31 All systems shall be capable of producing a consolidated report of the combination of overvotes for any contest that is selected by an authorized official (e.g.; the number of overvotes in a given contest combining candidate A and candidate B, combining candidate A and candidate C, etc.). {From VSS I.2.2.6.h and I.2.5.3.1.e}

4.2.2.16.32 All systems shall be capable of producing reports of all of the pre-election audit records, system readiness audit records, and in-process audit records defined in [xref Principle 8 audit record requirements]. {From VSS I.2.2.6.i, I.2.3.6 and I.2.5.3.1.f}

4.2.2.16.33 All systems shall provide the capabilities to obtain status and equipment readiness reports from each set of electronic equipment. {Reworded from VSS I.2.3.4.1.b}

[Discussion: ISSUE: status reports not defined.]

4.2.2.16.34 Systems claiming conformance to the *Unofficial results generation* profile shall provide only aggregated results in unofficial reports, and not data from individual ballots. {Reworded from VSS I.2.5.4a}

4.2.2.16.35 Systems claiming conformance to the *Unofficial results generation* profile shall clearly indicate on each unofficial report or file that the results it contains are unofficial. {Reworded from VSS I.2.5.4c}

4.2.2.16.36 All systems shall prevent data from being altered or destroyed by report generation, including data in transportable memory. {From VSS I.2.2.6.h, I.2.5.3.1.g, and I2.5.3.2d}

4.2.2.16.37 The interoperability requirement for report data may be met by providing the capability to export report data in a non-proprietary, open standard format. {Drill-down from TGDC Resolution #23-05}

4.2.2.16.38 The interoperability requirement for report data may be met by storing report data in a documented schema in a COTS or non-proprietary, open source database in such a manner that other applications can read and interpret the data. {Drill-down from VSS I.2.2.6 and Steve Freeman interpretation}

4.2.2.17 Conduct official audits

[Discussion: 2002 VSS I.2.2.5 defines general requirements for system audit. These should be detailed under Principle 2.8.]

4.2.2.17.1 All devices that tabulate ballots shall enable election officials to determine the number of ballots cast so far during a particular test cycle or election at any time during the test cycle or election without disrupting any operations in progress. {DWF, phrasing the functional requirement that was implied by design requirements in I.2.2.9}

[Discussion: 2002 VSS I.2.4 refers to separate "election counter" and "life-cycle counter;" the latter was an error (intended to delete).]

4.2.2.17.2 Systems claiming conformance to the *DRE* profile shall maintain an accurate Cast Vote Record of each ballot cast. {Reworded from VSS I.2.2.4.2}

4.2.2.17.3 Systems claiming conformance to the *DRE* profile shall provide a capability to retrieve ballot images in a form readable by humans. {VSS I.2.2.4.2.b and I.3.2.4.3.2.d}

4.2.2.17.4 All electronic systems shall provide software that monitors the overall quality of data read-write and transfer quality status, checking the number and types of errors that occur in any of the relevant operations on data and how they were corrected. {VSS I.2.2.2.1.e}

4.2.2.17.5 In systems claiming conformance to the *DRE* profile, the interoperability requirement for ballot image data may be met by providing the capability to export ballot image data in a non-proprietary, open standard format. {Drill-down from TGDC Resolution #23-05}

4.2.2.17.6 In systems claiming conformance to the *DRE* profile, the interoperability requirement for ballot image data may be met by storing ballot image data in a documented schema in a COTS or non-proprietary, open source database in such a manner that other applications can read and interpret the data. {Extrapolation from TGDC Resolution #23-05 and VSS I.2.2.6}

4.2.2.18 Procedural requirement

All printed copy records produced by the election database and ballot processing systems shall be labeled and archived for a period of at least 22 months after the election. {Reworded from VSS I.2.2.11}Responsible Entity: Voting Officials

4.2.2.19 Discussion: Design requirements of questionable worth

The following requirements constrain the design rather than specify the function or performance. There may be good reasons to constrain the design, and several justified design requirements have been retained. However, in the absence of rationale for constraining the design, the following will be deleted.

- All systems shall include control logic and data processing methods incorporating parity and check-sums (or equivalent error detection and correction methods) to demonstrate that the system has been designed for accuracy. {VSS I.2.2.2.1.d} – The presence of error detection and correction methods does not ensure that errors are actually detected and corrected. A tool is only effective if it is correctly used. 4.2.2.1.2 (end-to-end error rate) is more useful.

- The vote tabulating program software resident in each voting device, vote count server, or other devices shall include all software modules required to accumulate votes. {Reworded from VSS I.2.2.8.1} – Was the intent to prohibit dynamic loading of software? Not clear.

- Systems claiming conformance to the Precinct count profile shall provide a means to extract information from a transportable programmable memory device or data storage medium for vote consolidation. {Reworded from VSS I.2.5.3.2} – This is redundant – the reports must be generated somehow.

These ballot counter requirements are obsoleted by FR4.4.1. Steve Freeman believes that the assumption of a physical counter is a historical leftover from lever machines.

- All devices that tabulate ballots shall provide a counter that must be set to zero before any ballots are submitted for tally or DRE units are activated for voting. {Reworded from VSS I.2.2.9} <Testable> <Design>

- All devices that tabulate ballots shall provide a counter that records the number of ballots cast during a particular test cycle or election. {Reworded from VSS I.2.2.9} <Testable> <Design>

- All devices that tabulate ballots shall provide a counter that increases the count only by the acceptance of a cast ballot record. {Reworded from VSS I.2.2.9} <Testable> <Design>

- All devices that tabulate ballots shall provide a counter that prevents or disables the resetting of the counter by any person other than authorized persons at authorized points in the election cycle. {Reworded from VSS I.2.2.9} <Testable> <Design>

- All devices that tabulate ballots shall provide a counter that is visible to designated election officials. {Reworded from VSS I.2.2.9} <Testable> <Design>

4.2.2.20 Discussion: Requirements to be dealt with by Security & Transparency Subcommittee

- Systems claiming conformance to the *Unofficial results generation* profile shall provide no access path from unofficial electronic reports or files to the storage devices for official data. {Reworded from VSS I.2.5.4b} <Semi Testable> Refer to STS. (Access paths might not be obvious.)

- Systems claiming conformance to the *Precinct count* profile shall prevent the printing of reports and the unauthorized extraction of data prior to the official close of the polling place. {Reworded from VSS I.2.5.3.2} <Testable> Test by trying to do it / refer to STS.

- For systems claiming conformance to the *Central count* profile, the Voting Equipment User Documentation shall detail the measures to be taken related to the physical and procedural controls for handling of ballot boxes. {Reworded from VSS I.6.3.2} <Verifiable by inspection>

- For systems claiming conformance to the *Central count* profile, the Voting Equipment User Documentation shall detail the measures to be taken related to the physical and procedural controls for preparing of ballots for counting. {Reworded from VSS I.6.3.2} <Verifiable by inspection>

- For systems claiming conformance to the *Central count* profile, the Voting Equipment User Documentation shall detail the measures to be taken related to the physical and procedural controls for counting operations. {Reworded from VSS I.6.3.2} <Verifiable by inspection>

- For systems claiming conformance to the *Central count* profile, the Voting Equipment User Documentation shall detail the measures to be taken related to the physical and procedural controls for reporting data. {Reworded from VSS I.6.3.2} <Verifiable by inspection>

4.2.3 System Testing

4.2.3.1 Preface

This document contains test methods and testing standards related to the scope of work of the Core Requirements and Testing Subcommittee as of 2005-03-29. Other, very important test methods for security, usability, accessibility, and fitness for purpose (esp. environmental tests) are not contained here but are or will be distributed separately.

4.2.3.2 Data to be provided

[Technical Data Package]
(Changes / additions to current spec)

- An Implementation Statement, as defined in the Conformance Clause, including explicit statement of the capacities and limits within which the system is claimed to operate correctly.
- Source code, for systems using software; analogous formal logic designs, for systems not using software.
- For each distinct function, method, procedure, operation, etc., in source code or analogous logic design:
    - The preconditions and postconditions, formally stated using the terms defined in the Logic Verification section, including any assumptions about capacities and limits within which the system is expected to operate.[6]
    - A convincing argument (possibly, but not necessarily, a formal proof) that the preconditions and postconditions accurately represent the behavior of the function, method, procedure, operation, etc.[7]
    - A formal proof, using the preconditions and post-conditions, that the software or logic design as a whole satisfies each of the assertions indicated in the Assertions subsection for the profiles to which conformance is claimed, for all cases within the aforementioned capacities and limits.

4.2.3.3 Logic verification

Because of its high complexity, the scope of logic verification is necessarily limited to the core vote gathering and tabulating functions of specific components of the voting system (a voting machine and/or a central tabulator).

---

[6] The use of preconditions and postconditions as we have recommended first appeared in C. A. R. Hoare, "An Axiomatic Basis for Computer Programming," Communications of the ACM, v. 12, n. 10, October 1969, pp. 576-580, 583, with ideas derived from Robert W Floyd, "Assigning Meanings to Programs," in J. T. Schwartz, ed., Mathematical Aspects of Computer Science: Proceedings of Symposia in Applied Mathematics, v. 19, American Mathematical Society, 1967, pp. 19-32.

[7] Informality is permitted here to bridge the gap between a programming language with informal semantics and the formality that we require. The size limit on modules in source code (revised coding standards, v1s4, Sec. 4.2.3 b) is intended to keep modules small enough that preconditions and postconditions can be validated by inspection.

This model does not address ranked order voting and does not attempt to define every voting variation that jurisdictions may use. It suffices for 1 of M, N of M, and cumulative voting.

4.2.3.3.1 Domain of discourse

Preconditions and post-conditions shall be stated using the following terms:

Term	Definition
A(*t*,*v*)	Boolean function, returns true if and only if voter *v*'s ballot or ballot image conforms to jurisdiction-dependent criteria for accepting or rejecting entire ballots, such as stray marks policies and voter eligibility criteria, as of time *t*. This value is false for provisional, challenged, and review-required ballots that are not [yet] validated. The system may not be able to determine the value of A(*t*,*v*) without human input; however, it may assign tentative values according to local procedures and state law, to be corrected later if necessary by input from election workers. The value of A(*t*,*v*) may change over time as a result of court decisions, registrar review of voter eligibility, etc. In a paper-based system, A(*t*,*v*) will be false if voter *v*'s ballot is unprocessable.
B(*v*)	The time at which voter *v* begins voting (i.e., when the ballot is enabled).

C(r)	The set of all candidates or choices that are "on the ballot" in a contest r. Write-in candidates do not appear in C(r). *
C′(r,t)	The set of all candidates or choices for a contest r, including any write-ins that the voters have written in as of time t. Each distinct write-in candidate appears separately in C′(r,t). Where write-ins are not allowed, C′(r,t) = C(r). *
c, c_n, etc.	Individual candidates or choices.
D(v)	The time at which voter v is done voting (the time at which the ballot is cast or the ballot of a fleeing voter is spoiled).
J	The set of reporting contexts (including tabulators, precincts, election districts, and jurisdiction).
j, j_n, etc.	Individual reporting contexts.
K(j,r,t)	For a given contest and reporting context, the number of read ballots for which A(t,v) is true as of time t (i.e., the number of ballots that should be counted). Ballot forms that do not include contest r do not contribute to this total.
L_B	A limit on the number of ballots or ballot images that the system is claimed to be capable of processing correctly.
L_C	A limit on the number of ballot positions per contest that the system is claimed to be capable of processing correctly. (See also L_W)
L_F	A limit on the number of ballot forms that the system is claimed to be capable of processing correctly.
L_R	A limit on the number of contests that the system is claimed to be capable of processing correctly.
L_T	A numerical limit on vote totals that the system is claimed to be capable of processing correctly.
L_V	A limit on the number of voters casting provisional, challenged, or review-required ballots that the system is claimed to be capable of processing correctly.

L_W	A limit on the total number of distinct candidates or choices per contest, including write-ins, that the system is claimed to be capable of processing correctly. It shall be that $L_W \geq L_C$. (See also L_C)
$N(r)$	The maximum number of votes that may be cast by a given voter in contest *r*, pursuant to the definition of the contest. For N of M contests, this is the value N.
$O(j,r,t)$	For a given contest and reporting context, the number of overvotes in read ballots for which $A(t,v)$ is true as of time *t*. Each ballot in which contest *r* is overvoted contributes $N(r)$ to $O(j,r,t)$.
R	The set of all contests.
r, r_n, etc.	Individual contests in R.
$S(c,r,t,v)$	Voter *v*'s vote with respect to candidate or choice *c* in contest *r* as of time *t*. For checkboxes and the like, the value shall be 1 (selected) or 0 (not selected). For cumulative voting, the value shall be the number of votes that *v* gives to candidate or choice *c* in contest *r*. If the applicable ballot form does not include contest *r*, $S(c,r,t,v) = 0$.
$S'(c,r,t,v)$	Voter *v*'s vote with respect to candidate or choice *c* in contest *r* as accepted for counting purposes (i.e., valid votes only), as of time *t*.
$S(r,t,v)$	The total number of votes that voter *v* has cast in contest *r* as of time *t*, $= \sum_{c \in C'(r,t)} S(c,r,t,v)$
$T(c,j,r,t)$	The vote total for candidate or choice *c* in contest *r* and reporting context *j* as of time *t*. This does not include votes that are invalid due to overvoting or votes from ballots for which $A(t,v)$ is false.
t, t_n, etc.	Individual time points.
t_O	The time at which polls are opened.
t_C	The time at which polls are closed.

t_E	The time at which the value of A(*t*,*v*) is frozen for all voters, the counting is complete, and final vote totals are required ("end").
U(*j*,*r*,*t*)	For a given contest and reporting context, the number of undervotes in read ballots for which A(*t*,*v*) is true as of time *t*. A given ballot contributes at most N(*r*) to U(*j*,*r*,*t*). Ballot forms that do not include contest *r* do not contribute to this total.
V(*j*,*t*)	The set of all voters within reporting context *j* who have begun voting by time *t*, including any voter that is presently voting.
v, v_n, etc.	Individual voters in V(*j*,*t*).

* The fact that some systems initially report "Write-In" as a single ballot position, leaving the distribution of votes to different write-in candidates for post-processing, is an implementation detail. These standards contain requirements on the information content of the final report, which must provide separate totals for each write-in candidate.

The scope of these terms is herein referred to as the domain of discourse. Post-conditions that impact something outside the domain of discourse are not of interest unless that thing impacts the behavior of some function with respect to the domain of discourse. The vendor shall define such terms as are necessary to state any and all dependencies and assumptions that may impact the behavior of some function with respect to the domain of discourse and use them consistently in all affected preconditions and post-conditions. *An excess of extraneous dependencies may negatively impact the VSTL's ability to determine the system's correctness and thereby prevent qualification.* A function may have no impact on anything in the domain of discourse and no dependency on anything in the domain of discourse. Such a function shall have a true precondition and a post-condition that states that nothing in the domain of discourse is changed.

[Discussion: Possibly "voters" should be replaced with "ballots" in as many places as possible in this model to avoid suggesting a loss of privacy. However, the need remains to maintain 1-voter to 1-ballot parity somehow.]

4.2.3.3.2 Assertions

General invariants:

$t_O < t_C \leq t_E$

$v \in V(j,t) \rightarrow B(v) \leq t$

$B(v) < D(v)$

$S(c,r,t,v) \geq 0$

$S'(c,r,t,v) \geq 0$

$S(c,r,t,v) > 0 \rightarrow c \in C'(r,t)$

The following assertions formalize a subset of the compliance points appearing in Vol. II. Each textual assertion is intended to elucidate the formal assertion(s) that follow it. In case of discrepancy or confusion, the formal assertions are normative.

No one shall vote before polls are opened or after polls have closed, or during the process of opening or closing the polls.

$B(v) > t_O$

$D(v) < t_C$

A voter shall have no votes before he or she begins voting.

$t < B(v) \rightarrow S(r,t,v) = 0$

A voter's votes shall not change once the voter is done voting.

$t \geq D(v) \rightarrow S(c,r,t,v) = S(c,r,D(v),v)$

4.2.3.3.2.1 Cumulative voting

All valid votes shall be counted.

$t \geq D(v) \wedge S(r,D(v),v) \leq N(r) \wedge A(t,v) \rightarrow S'(c,r,t,v) = S(c,r,D(v),v)$

No invalid votes shall be counted.

$t \geq D(v) \wedge (S(r,D(v),v) > N(r) \vee \sim A(t,v)) \rightarrow S'(c,r,t,v) = 0$

The final vote totals shall accurately reflect all valid votes and only valid votes.

$$T(c, j, r, t_E) = \sum_{v \in V(j, t_E)} S'(c, r, t_E, v)$$

Every vote shall be accounted for.

$$\sum_{c \in C'(r,t_E)} T(c, j, r, t_E) + O(j,r,t_E) + U(j,r,t_E) = K(j,r,t_E) \times N(r)$$

4.2.3.3.2.2 N of M contests (including 1-of-M)

N of M is identical to cumulative voting but for the addition of the following invariant, which reflects the design of a ballot form that allows only one vote in each ballot position (equivalent to a checkbox).

$$S(c,r,t,v) \leq 1$$

4.2.3.3.3 Reporting

The phrase "shall publish" indicates information that shall appear in the Public Information Package as well as the Qualification Test Report. The phrase "shall report" indicates information that shall appear in the Qualification Test Report. The term "finding" refers to a result of the VSTL's formal inquiry (a verdict).[8]

For each distinct function, method, procedure, operation, etc., in source code or analogous logic design, the VSTL shall publish a finding on whether the preconditions and post-conditions correctly describe the behavior of the function in all cases. This finding shall be one of Correct, Incorrect, or Unable to Determine. No system shall be qualified unless all preconditions and post-conditions are found Correct.

The VSTL shall publish a finding whether the assumptions about capacities and limits that appear in the preconditions, post-conditions, and proofs are consistent with the capacities and limits that the system is claimed to be capable of processing correctly. This finding shall be one of Consistent, Inconsistent, or Unable to Determine. No system shall be qualified unless the assumptions about capacities and limits are found Consistent.

For the software or logic design as a whole, and for each assertion indicated above for the voting variation profiles to which conformance is claimed, the VSTL shall publish a finding whether the assertion is satisfied in all cases within the aforementioned capacities and limits. This finding shall be one of Satisfied, Unsatisfied, or Unable to Determine. No system shall be qualified unless all assertions are found Satisfied.

4.2.3.4 Design requirement verification

For each of the design requirements enumerated below, the VSTL shall review the source code (if applicable) and design of the voting system to verify that the requirement is satisfied. For each one, the VSTL shall publish a finding whether the requirement is met. This finding shall be one of

[8] Based on finding, definition 6, in the New Shorter Oxford English Dictionary, 1993.

Satisfied, Unsatisfied, or Unable to Determine. No system shall be qualified unless all design requirements are found Satisfied.

4.2.3.4.1 All systems shall maintain the integrity of voting and audit data, including Cast Vote Records, during an election and for a period of at least 22 months afterward.[9]

4.2.3.4.2 All systems shall maximize interoperability and integratability with other systems and/or components of other systems.[10]

4.2.3.4.3 Systems claiming conformance to the *DRE* profile shall verify (i.e., actively check and confirm) the correct addition of voter selections to the memory components of the device.

4.2.3.4.4 All electronic systems shall verify (i.e., actively check and confirm) that information as extracted or duplicated to machine-readable media is identical to that on the original storage medium.

4.2.3.4.5 Transmitting results over telecommunications lines shall not result in modifications to any vote data in the sending system.

4.2.3.4.6 All electronic systems shall ensure that information received by transmission over telecommunications lines is identical to what was sent.

4.2.3.4.7 All systems shall prevent data from being altered or destroyed by report generation, including data in transportable memory.

4.2.3.5 General test template

Most test cases will follow this general template. Different test cases will elaborate on the general template in different ways, depending on what is being tested.

1. Establish initial state (clean out data from previous tests, verify resident software/firmware)
2. Program election and prepare ballots
3. Generate pre-election audit reports
4. Configure polling equipment
5. Generate system readiness audit reports
6. Open poll
7. Run test ballots
8. Close poll
9. Generate in-process audit reports

[9] The VSTL should rely on authoritative information regarding the archivalness of various storage media.

[10] Sub-requirements FR4.6.1 through FR4.6.6 indicate acceptable designs.

10. Generate data reports for the specified reporting contexts
11. Inspect ballot counters
12. Inspect reports

4.2.3.6 General pass criteria

The VSTL need only consider tests that are applicable to the profiles claimed in the Implementation Statement and those tests that are designated for all systems. The test verdict for all other tests shall be Not Applicable.

If the documented assumptions for a given test (indicated by the presence of an **Assumptions:** field in the test case description) are not met, the test verdict shall be Waived and the test shall not be executed.

If the VSTL is unable to execute a given test because the system does not support functionality that is required per the Implementation Statement or is required for all systems, the test verdict shall be Fail.

If the VSTL executes a test, the test verdict shall be assigned based on the following inputs, which are described in more detail below:

- Mean Time Between Failure (MTBF)
- Error rate
- Additional pass criteria
- General performance requirements

The test verdict shall be Pass if and only if none of these inputs indicates a verdict of Fail. No system shall be qualified if any test verdicts are Fail.

4.2.3.6.1 Mean Time Between Failure

During execution of all tests except *xref (recovery with forced errors)*, the VSTL shall keep track of real time and the number of operational failures. These statistics shall be collected and accumulated across all tests.

An operational failure is defined as any event that results in (a) loss of one or more functions, (b) degradation of performance such that the device is unable to perform its intended function for longer than 10 seconds, (c) automatic reset, restart or reboot of the voting system, operating system or application software, (d) a requirement for intervention by a person in the role of poll worker or technician before the test can continue, or (e) error messages and/or audit log entries indicating that a failure has occurred.

If an operational failure should occur during the execution of any test except *xref (recovery with forced errors)*, the VSTL shall note the failure for use in the calculation of MTBF. The VSTL shall then follow the vendor's documented procedures for recovering from operational failures. If

recovery is not possible or not successful, the test verdict shall be Fail. Otherwise, after recovery, the VSTL shall attempt to re-execute the test that was affected by the operational failure from the beginning. If the failure reoccurs, the test verdict shall be Fail. If the failure does not reoccur, the following system-level MTBF decision criteria shall be applied:

- If statistical analysis of the cumulative behavior across all tests executed so far indicates with at least 95% confidence that the MTBF is worse than 500 hours, the test verdict shall be Fail.
- Otherwise, the failure shall be noted, the test verdict shall be assigned based on the other inputs (disregarding the operational failure), and testing shall continue.

Discussion: Definition of operational failure is expanded from definition in 2002 VSS I.3.4.3. 500 hour MTBF is tentative, arbitrary revision of 163-hour requirement, pending formal analysis by statistics team to determine correct setting.

4.2.3.6.2 Error rate

During all test executions, the VSTL shall keep track of the number of ballot positions counted and the number of errors (ballot positions counted incorrectly). These statistics shall be collected and accumulated across all tests.

If a test runs to completion, the VSTL shall inspect the data reports and verify that counts and totals are reported in compliance with the requirements in Vol. II, Principle 4, A1.8.4 (Report). If all reported counts and totals are identical to the specified values, the test verdict shall be Pass. Otherwise, the following system-level accuracy decision criteria shall be applied:

- If statistical analysis of the cumulative behavior across all tests executed so far indicates with at least 95% confidence that the error rate is worse than 1 in 10,000,000 ballot positions, the test verdict shall be Fail.
- Otherwise, the failure shall be noted, the test verdict shall be assigned based on the other inputs (disregarding the errors), and testing shall continue.

4.2.3.6.3 Additional pass criteria

When certain performance requirements of the VVSG are of particular relevance to a particular test, these are noted after **Additional pass criteria:** in the test case description. The VSTL shall verify that these requirements are met during the execution of that test case; if they are not, the test verdict shall be Fail.

4.2.3.6.4 General performance requirements

A demonstrable violation of any requirement of the VVSG during the execution of any test case shall result in a test verdict of Fail, irrespective of whether this requirement was explicitly noted in the Additional pass criteria for that test case.

For example, if any of the audit reports should be incomplete or incorrect with respect to any of the many applicable requirements in *(xref Principle 2.8, 2002 VSS I.4.4*), the test verdict would be Fail.

For example, if a DRE system should take longer than 5 minutes for each device to generate a consolidated report, FR4.3.3 would be violated and the test verdict would be Fail.

4.2.3.7 General reporting requirements

If a system is qualified, the VSTL shall publish a statement to that effect that includes the system identification, the profiles claimed in the Implementation Statement, the assumptions about capacities and limits, a list of the tests for which the test verdict was Waived, and the estimated error rate and MTBF of the system as calculated from the statistics collected during testing. For systems claiming conformance to the *Optical Scan* or *Punchcard* profiles, the VSTL shall also publish the speed or rate at which tabulation was performed in typical case and capacity tests.

Whether or not a system is qualified, the VSTL shall report all of the data collected for estimation of MTBF and error rate.

If a system is not qualified, the VSTL shall report on all failed tests and the reasons for failure, including all applicable evidence (e.g., vote data report, proof of logic error in source code).

4.2.3.8 Null Case Test

The purpose of the null case test is to verify that closing the polls after processing zero ballots is correctly handled. This case can arise in practice, for example, in precincts where a single DRE is provided alongside other equipment, if no voters use the DRE.

4.2.3.8.1 All systems

2.4.5.8.1.1 Test case name: Null Case

Ballot form: **1** 1-of-M contest where M = 1.
The contest shall be described as follows:
This is the only contest in the Null Case Test. There is only one candidate on the ballot.
The only ballot position in the contest shall be the following:
Unopposed Candidate
Reporting contexts: Single context.
Scenario: No ballots shall be cast.

4.2.3.9 Functional tests

The purpose of a functional test is to establish that one or more functional features that are required to be supported, are supported. Functional tests are not stress tests, although by their

minimalism, they may unintentionally test boundary conditions. For stress tests, refer to the Capacity Tests section.

Following subsections are organized by compliance profiles. Functional tests are applicable only if the Implementation Statement asserts conformance to the profile indicated in the subsection name.

4.2.3.9.1 All systems

4.2.3.9.1.1 **Test case name:** 1-of-M Trivial Case

Ballot form: 1 1-of-M contest where M = 1.
The contest shall be described as follows:
This is the only contest in the 1-of-M Trivial Case Test. There is only one candidate on the ballot.
The only ballot position in the contest shall be the following:
Unopposed Candidate
Reporting contexts: Single context.
Scenario: Two ballots shall vote for Unopposed Candidate.

4.2.3.9.1.2 **Test case name:** 1-of-M Simple Case

Ballot form: 1 1-of-M contest where M = 3.
The contest shall be described as follows:
This is the only contest in the 1-of-M Simple Case Test. There are three candidates on the ballot. Vote for at most one.
The ballot positions shall be the following:
Ballot Position 1
Ballot Position 2
Ballot Position 3
Reporting contexts: Single context.
Scenario: Four ballots shall vote for Ballot Position 1
Three ballots shall vote for Ballot Position 2
Two ballots shall vote for Ballot Position 3
One ballot shall vote for none (undervote).

4.2.3.9.1.3 **Test case name:** Reporting Levels Test

Six voting machines, three precinct tabulators and one jurisdiction tabulator are required to execute this test.

Ballot form: 1 1-of-M contest where M = 3.
The contest shall be described as follows:
This is the only contest in the Reporting Levels Test. There are three candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:
Ballot Position 1
Ballot Position 2
Ballot Position 3

Reporting contexts: Machines 1 and 2 shall be in Precinct 1.
Machines 3 and 4 shall be in Precinct 2.
Machines 5 and 6 shall be in Precinct 3.
Precincts 1 and 2 shall be in District 1.
Precinct 3 shall be in District 2.

All of the above shall be in the Jurisdiction.

Scenario:

- On Machine 1, three ballots shall be cast for Ballot Position 1, two ballots shall be cast for Ballot Position 2, and one ballot shall be cast for Ballot Position 3.
- On Machine 2, three ballots shall be cast for Ballot Position 1, one ballot shall be cast for Ballot Position 2, and one ballot shall be cast for Ballot Position 3.
- On Machine 3, two ballots shall be cast for Ballot Position 1, one ballot shall be cast for Ballot Position 2, and one ballot shall be cast for Ballot Position 3.
- On Machine 4, one ballot shall be cast for Ballot Position 1, one ballot shall be cast for Ballot Position 2, and one ballot shall be cast for Ballot Position 3.
- On Machine 5, two ballots shall be cast for Ballot Position 2.
- On Machine 6, one ballot shall be cast for Ballot Position 1.

4.2.3.9.2 DRE

4.2.3.9.2.1 **Test case name:** Ballot Images Simple Case

Ballot form: 1 1-of-M contest where M = 3.

The contest shall be described as follows:
This is the only contest in the Ballot Images Simple Case Test. There are three candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:
Ballot Position 1
Ballot Position 2
Ballot Position 3

Reporting contexts: Single context.

Scenario:

- Four ballots shall vote for Ballot Position 1
- Three ballots shall vote for Ballot Position 2
- Two ballots shall vote for Ballot Position 3
- One ballot shall vote for none (undervote).
- After close of polls, the VSTL shall retrieve and review the ballot images.

Additional pass criteria:

- The Cast Vote Records (retrieved ballot images) shall be accurate. (FR4.4.2)
- The Cast Vote Records (retrieved ballot images) shall be human-readable. (FR4.4.3)

- The Cast Vote Records (retrieved ballot images) shall not be reported in the same order in which they were voted. *(xref privacy requirement, from 2002 VSS I.3.2.4.3.2.e*)

4.2.3.9.3 Optical Scan and Punchcard

4.2.3.9.3.1 **Test case name:** Overvoting Simple Case

Ballot form: 1 1-of-M contest where M = 3.
The contest shall be described as follows:
This is the only contest in the Overvoting Simple Case Test. There are three candidates on the ballot. Vote for at most one.
The ballot positions shall be the following:
Ballot Position 1
Ballot Position 2
Ballot Position 3
Reporting contexts: Single context.
Scenario:
- Three ballots shall vote for Ballot Position 1
- Two ballots shall vote for Ballot Position 1 and Ballot Position 2
- Two ballots shall vote for Ballot Position 2
- Three ballots shall vote for Ballot Position 2 and Ballot Position 3
- One ballot shall vote for Ballot Position 3
- Four ballots shall vote for Ballot Position 1 and Ballot Position 3
- One ballot shall vote for all three ballot positions
- One ballot shall vote for none (undervote).
- In addition to generating the usual reports, the VSTL shall perform four ad-hoc queries to determine the number of overvotes combining ballot positions in each of the four applicable combinations (1+2, 1+3, 2+3, 1+2+3).

4.2.3.9.4 Closed primaries

4.2.3.9.4.1 **Test case name:** Closed Primary Simple Case

Ballot form: Whig 2 1-of-M contests where M = 2.
The first contest shall be described as follows:
This is the first contest in the Whig ballot form of the Closed Primary Simple Case Test. There are two candidates on the ballot. Vote for at most one.
The ballot positions shall be the following:
Whig 1
Whig 2
The second contest shall be described as follows:
This is the second contest, a non-partisan office. There are two candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

Whig 3
Tory 3

Ballot form: Tory 2 1-of-M contests where M = 2.
The first contest shall be described as follows:

This is the first contest in the Tory ballot form of the Closed Primary Simple Case Test. There are two candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

Tory 1
Tory 2

The second contest (non-partisan) shall be identical to the second contest in the Whig ballot form.
Reporting contexts: Single context.
Scenario:

- Two Whig ballots shall vote for Whig 1 and Whig 3
- One Whig ballot shall vote for Whig 2 and Tory 3
- One Tory ballot shall vote for Tory 1 and Tory 3
- Two Tory ballots shall vote for Tory 2 and skip the second contest (undervotes).

Additional pass criteria: Separate counts for each party shall be reported in accordance with FR4.3.6.2 and FR4.3.7.1.

4.2.3.9.5 Open primaries

4.2.3.9.5.1 Test case name: Open Primary Simple Case

Ballot forms: Ballot forms shall be identical to those in Closed Primary Simple Case, except changing the name of the test case in the contest descriptions.
Reporting contexts: Single context.
Scenario: same as Closed Primary Simple Case.
Additional pass criteria: Separate counts for each party shall be reported in accordance with FR4.3.6.2 and FR4.3.7.1.
The voter shall be allowed to choose a party affiliation at the time of voting and vote the appropriate ballot form in accordance with FR4.1.4.2.1.

4.2.3.9.6 Write-ins

4.2.3.9.6.1 Test case name: Write-ins Simple Case

Ballot form: 1 1-of-M contest where M = 1.
The contest shall be described as follows:

This is the only contest in the Write-ins Simple Case Test. There are no candidates on the ballot. Write in at most one.

The only ballot position in the contest shall be a write-in opportunity.

Reporting contexts: Single context.
Scenario:
- Four ballots shall write in First Write-In Candidate.
- Three ballots shall vote for none (undervote).
- Two ballots shall write in Second Write-In Candidate.

4.2.3.9.7 Ballot rotation

4.2.3.9.7.1 **Test case name:** Ballot Rotation Simple Case

Ballot form: 1 1-of-M contest where M = 3, with ballot rotation enabled.
The contest shall be described as follows:

> This is the only contest in the Ballot Rotation Simple Case Test. There are three candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

> Candidate 1
> Candidate 2
> Candidate 3

Reporting contexts: Single context.
Scenario:
- Four ballots shall vote for Candidate 1
- Three ballots shall vote for Candidate 2
- Two ballots shall vote for Candidate 3

Additional pass criteria: Each candidate shall appear in each position on the ballot exactly three times in accordance with 4.1.4.4.1.

4.2.3.9.8 Straight party voting

4.2.3.9.8.1 **Test case name:** Straight Party Voting Simple Case

Ballot form: 2 1-of-M contests.
The first contest shall be described as follows:

> STRAIGHT PARTY. If you desire to vote a straight party ticket for all offices, vote for at most one party here. Votes for individual candidates in subsequent contests will override the straight party vote in those contests only.

The ballot positions shall be the following:

> Whig
> Tory

The second contest shall be described as follows:

> This is the only contest in the Straight Party Voting Simple Case Test. There are three candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

> Ballot Position 1 (Whig)
> Ballot Position 2 (Tory)
> Ballot Position 3 (Independent)

Reporting contexts: Single context.
Scenario:

- Two ballots shall vote straight party Whig and skip the second contest (allowing the straight party vote to be effective)
- One ballot shall skip the straight party vote and vote for Ballot Position 1 (Whig)
- One ballot shall votes straight party Tory but then vote for Ballot Position 1 (Whig)
- Two ballots shall vote straight party Tory and skip the second contest
- One ballot shall skip the straight party vote and vote for Ballot Position 2 (Tory)
- One ballot shall vote straight party Tory but then vote for Ballot Position 3 (Independent).
- (Expected result: Ballot Position 1 (Whig), 4; Ballot Position 2 (Tory), 3; Ballot Position 3 (Independent), 1.)

4.2.3.9.9 Cross-party endorsement

4.2.3.9.9.1 **Test case name:** Cross-party Endorsement Simple Case

Ballot form: 2 1-of-M contests.
The first contest shall be described as follows:

> STRAIGHT PARTY. If you desire to vote a straight party ticket for all offices, vote for at most one party here. Votes for individual candidates in subsequent contests will override the straight party vote in those contests only.

The ballot positions shall be the following:

> Whig
> Free-Soil
> National
> Federalist

The second contest shall be described as follows:

> This is the only contest in the Straight Party Voting Simple Case Test. There are three candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

> Ballot Position 1 (Whig/National/Federalist)
> Ballot Position 2 (Free-Soil)
> Ballot Position 3 (Independent)

Reporting contexts: Single context.
Scenario:

- One ballot shall vote straight party Whig and skip the second contest
- Two ballots shall vote straight party Free-Soil and skip the second contest
- Three ballots shall vote straight party National and skip the second contest
- Two ballots shall vote straight party Federalist and skip the second contest

- One ballot shall skip the straight party vote and vote for Ballot Position 2 (Free-Soil)
- One ballot shall skip the straight party vote and vote for Ballot Position 3 (Independent).

4.2.3.9.10 Split precincts

4.2.3.9.10.1 **Test case name:** Split Precinct Simple Case

Ballot form: District 1: 1 1-of-M contest where M = 2.
The contest shall be described as follows:

> This is the only contest in the District 1 ballot form of the Split Precinct Simple Case Test. There are two candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

> District 1 Candidate 1
> District 1 Candidate 2

Ballot form: District 2: 1 1-of-M contest where M = 2.
The contest shall be described as follows:

> This is the only contest in the District 2 ballot form of the Split Precinct Simple Case Test. There are two candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

> District 2 Candidate 1
> District 2 Candidate 2

Reporting contexts: Precinct 1, District 1, District 2, Jurisdiction. (Precinct 1 is split between District 1 and District 2.)
Scenario:

- Three District 1 ballots shall vote for District 1 Candidate 1
- Two District 1 ballots shall vote for District 1 Candidate 2
- Six District 2 ballots shall vote for District 2 Candidate 1

Additional pass criteria:

- Only the District 1 ballots shall be reported in the District 1 reporting context.
- Only the District 2 ballots shall be reported in the District 2 reporting context.
- All ballots shall be reported in the Precinct 1 and Jurisdiction reporting contexts.

4.2.3.9.11 N of M voting

4.2.3.9.11.1 **Test case name:** N-of-M Simple Case

Ballot form: 1 N-of-M contest where N = 2 and M = 3.
The contest shall be described as follows:

> This is the only contest in the N-of-M Simple Case Test. There are three candidates on the ballot. Vote for at most two.

The ballot positions shall be the following:

Ballot Position 1
Ballot Position 2
Ballot Position 3

Reporting contexts: Single context.

Scenario:

- Four ballots shall vote for Ballot Position 1 and Ballot Position 2
- Three ballots shall vote for Ballot Position 1 and Ballot Position 3
- Two ballots shall vote for Ballot Position 2 and nobody else (single undervote)
- One ballot shall vote for none (double undervote).

4.2.3.9.12 N of M voting + Write-ins

4.2.3.9.12.1 **Test case name:** N-of-M + Write-ins Simple Case

Ballot form: 1 N-of-M contest where N = 2 and M = 3.

The contest shall be described as follows:

This is the only contest in the N-of-M + Write-ins Simple Case Test. There are no candidates on the ballot. Write in at most two.

The ballot positions shall be two write-in opportunities.

Reporting contexts: Single context.

Scenario:

- Four ballots shall write in "Write-in Candidate 1" and "Write-in Candidate 2"
- Two ballots shall write in "Write-in Candidate 1" and "Write-in Candidate 3"
- Four ballots shall write in "Write-in Candidate 2" and nobody else (single undervote)
- One ballot shall vote for none (double undervote).

4.2.3.9.13 Cumulative voting

4.2.3.9.13.1 **Test case name:** Cumulative Voting Simple Case

Ballot form: 1 cumulative voting contest where M = 3 and N(*r*) = 3.

The contest shall be described as follows:

This is the only contest in the Cumulative Voting Simple Case Test. There are three candidates on the ballot. Cast at most three votes. You may cast multiple votes for the same candidate.

The ballot positions shall be the following:

Ballot Position 1
Ballot Position 2
Ballot Position 3

Reporting contexts: Single context.

Scenario:

- Four ballots shall vote once each for Ballot Position 1, Ballot Position 2, and Ballot Position 3

- Three ballots shall vote twice for Ballot Position 2 and once for Ballot Position 3
- Two ballots shall vote three times for Ballot Position 3
- One ballot shall vote for none (undervote).

4.2.3.9.14 Ranked order voting

4.2.3.9.14.1 **Test case name:** Ranked Order Voting Simple Case

Ballot form: 1 1-of-M ranked order contest where M = 3.
The contest shall be described as follows:
This is the only contest in the Ranked Order Voting Simple Case Test. There are three candidates on the ballot. Please rank them in order of preference.
The ballot positions shall be the following:
Ballot Position 1
Ballot Position 2
Ballot Position 3
Reporting contexts: Single context.
Scenario:
- Four ballots vote shall the ordering Ballot Position 1, Ballot Position 2, Ballot Position 3
- Three ballots shall vote the ordering Ballot Position 2, Ballot Position 3, Ballot Position 1
- Two ballots shall vote the ordering Ballot Position 3, Ballot Position 2, Ballot Position 1.
- (Expected result:
  Round 1: Ballot Position 1, 4
  Ballot Position 2, 3
  Ballot Position 3, 2
  Round 2: Ballot Position 2, 5
  Ballot Position 1, 4)

4.2.3.9.15 Provisional / challenged ballots

4.2.3.9.15.1 **Test case name:** Provisional Ballots Simple Case

Ballot form: 1 1-of-M contest where M = 3.
The contest shall be described as follows:
This is the only contest in the Provisional Ballots Simple Case Test. There are three candidates on the ballot. Vote for at most one.
The ballot positions shall be the following:
Ballot Position 1
Ballot Position 2
Ballot Position 3
Reporting contexts: Single context.

Scenario:
- Two regular ballots shall vote for Ballot Position 1
- Five provisional ballots shall vote for Ballot Position 1, and two shall be accepted
- Three regular ballots shall vote for Ballot Position 2
- One regular ballot shall vote for Ballot Position 3
- Four provisional ballots shall vote for Ballot Position 3, and one shall be accepted
- One provisional ballot shall vote for none (undervote), and shall be accepted.

Additional pass criteria: The number of provisional / challenged ballots cast, read and counted shall be reported in compliance with FR4.3.5.1, FR4.3.6.3, and FR4.3.7.2.

4.2.3.9.16 Unofficial results generation

4.2.3.9.16.1 Test case name: Unofficial Results Simple Case

Ballot form: 1 1-of-M contest where M = 3.
The contest shall be described as follows:

This is the only contest in the Unofficial Results Simple Case Test. There are three candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

Ballot Position 1
Ballot Position 2
Ballot Position 3

Reporting contexts: Single context.
Scenario:
- Four ballots shall vote for Ballot Position 1
- Three ballots shall vote for Ballot Position 2
- Two ballots shall vote for Ballot Position 3
- One ballot shall vote for none (undervote).
- In addition to the usual reports specified in the general test template, an unofficial vote data report shall be generated.

Additional pass criteria:
- The unofficial report shall provide only aggregated results in unofficial reports, and not data from individual ballots. (FR4.3.16)
- The unofficial report shall clearly indicate that the results it contains are unofficial. (FR4.3.17)

4.2.3.10 Typical case tests

The purpose of typical case tests is to test the behavior of the voting system in scenarios that reflect typical use of the system in practice rather than artificial minimum and maximum conditions.

Special instructions for Optical Scan and Punchcard: For systems claiming conformance to the *Optical Scan* or *Punchcard* profiles, all applicable typical case tests shall be executed at a tabulating rate no less than 30 ballots per minute, or the maximum rate at which the tabulating equipment is documented to function reliably, whichever is less. To speed testing, a higher rate may be used if the vendor does not object.

[Discussion: Default tabulating rate is from 1990 VSS J-3.]

4.2.3.10.1 All systems

4.2.3.10.1.1 **Test case name:** 1-of-M Typical Case

Assumptions:

$L_R \geq 10$
$L_C \geq 10$
$L_B \geq 75000$
$L_T \geq 38375$

Ballot form: 10 1-of-M contests where M = 10.

- The contests shall be described as follows (substituting numbers from 1 to 10 for *r*):
- This is Contest *r* in the 1-of-M Typical Case Test. Vote for at most one.
- The ballot positions in each contest shall be of the following form (substituting numbers from 1 to 10 for *c*):
- Contest *r* Ballot Position *c*

There are no write-in ballot positions in this ballot form.

Reporting contexts: Precinct 1, Precinct 2, … through Precinct 100, Jurisdiction.

Scenario: A total of 75000 ballots shall be cast. Precincts 1 through 50 shall each receive 750 – *n* ballots, for precinct number *n*. Precincts 51 through 100 shall each receive 700 + *n* ballots.

- In all precincts,
  345 ballots shall vote for ballot position 1 in every contest
  345 ballots shall vote for ballot position 2 in every contest
  1 ballot shall vote for ballot position 3 in contest 3 and undervote the rest
  1 ballot shall vote for ballot position 4 in contest 4 and undervote the rest
  1 ballot shall vote for ballot position 5 in contest 5 and undervote the rest
  1 ballot shall vote for ballot position 6 in contest 6 and undervote the rest
  1 ballot shall vote for ballot position 7 in contest 7 and undervote the rest
  1 ballot shall vote for ballot position 8 in contest 8 and undervote the rest
  1 ballot shall vote for ballot position 9 in contest 9 and undervote the rest
  1 ballot shall vote for ballot position 10 in contest 10 and undervote the rest
- In precincts 1 through 50, all remaining ballots shall vote for ballot position 1 in the first contest and undervote the rest.
- In precincts 51 through 100, all remaining ballots shall vote for ballot position 2 in the first contest and undervote the rest.

[Discussion: This test is based loosely on the minimum acceptance test guidelines in Appendix J of the 1990 Voting Systems Standards. It has been modified to remove the explicit requirement for a large number of voting machines for testing – such are unlikely to be available for a system that is not yet qualified. The requirement for N-of-M voting has been removed to permit the test to apply to all systems.]

4.2.3.10.2 Optical Scan and Punchcard

4.2.3.10.2.1 **Test case name:** 1-of-M Paper Typical Case

Assumptions:

$L_R \geq 10$
$L_C \geq 10$
$L_B \geq 75000$
$L_T \geq 34500$

Ballot form:10 1-of-M contests where M = 10.

The contests shall be described as follows (substituting numbers from 1 to 10 for *r*):

This is Contest *r* in the 1-of-M Paper Typical Case Test. Vote for at most one.

The ballot positions in each contest shall be of the following form (substituting numbers from 1 to 10 for *c*):

Contest *r* Ballot Position *c*

There are no write-in ballot positions in this ballot form.

Reporting contexts: Precinct 1, Precinct 2, … through Precinct 100, Jurisdiction.

Scenario: A total of 75000 ballots shall be cast. Precincts 1 through 50 shall each receive 750 – *n* ballots, for precinct number *n*. Precincts 51 through 100 shall each receive 700 + *n* ballots.

- In all precincts,
  345 ballots shall vote for ballot position 1 in every contest
  345 ballots shall vote for ballot position 2 in every contest
  1 ballot shall vote for ballot position 3 in contest 3 and undervote the rest
  1 ballot shall vote for ballot position 4 in contest 4 and undervote the rest
  1 ballot shall vote for ballot position 5 in contest 5 and undervote the rest
  1 ballot shall vote for ballot position 6 in contest 6 and undervote the rest
  1 ballot shall vote for ballot position 7 in contest 7 and undervote the rest
  1 ballot shall vote for ballot position 8 in contest 8 and undervote the rest
  1 ballot shall vote for ballot position 9 in contest 9 and undervote the rest
  1 ballot shall vote for ballot position 10 in contest 10 and undervote the rest
- In precincts 1 through 50, all remaining ballots shall overvote the first contest by voting for both ballot positions 1 and 3 and undervote the rest.
- In precincts 51 through 100, all remaining ballots shall overvote the first contest by voting for both ballot positions 2 and 3 and undervote the rest.

4.2.3.10.3 N of M voting

4.2.3.10.3.1 **Test case name:** N-of-M Typical Case

Assumptions:

$L_R \geq 10$
$L_C \geq 10$
$L_B \geq 75000$
$L_T \geq 75000$

Ballot form:

There shall be 10 contests. The first contest shall be an N-of-M contest where M = N = 10.

The first contest shall be described as follows:

This is Contest 1 in the N-of-M Typical Case Test. Vote for at most 10.

The other 9 contests shall be 1-of-M contests where M = 10. These contests shall be described as follows (substituting numbers from 2 to 10 for *r*):

This is Contest *r* in the N-of-M Typical Case Test. Vote for at most one.

The ballot positions in all 10 contests shall be of the following form (substituting numbers from 1 to 10 for *c*):

Contest *r* Ballot Position *c*

There are no write-in ballot positions in this ballot form.

Reporting contexts: Precinct 1, Precinct 2, … through Precinct 100, Jurisdiction.

Scenario: A total of 75000 ballots shall be cast. Precincts 1 through 50 shall each receive 750 – *n* ballots, for precinct number *n*. Precincts 51 through 100 shall each receive 700 + *n* ballots.

- In precincts 1 through 50, all ballots shall vote for all 10 candidates in Contest 1.
- In precincts 51 through 100, all ballots shall vote for only the first 8 ballot positions in Contest 1 (yielding two undervotes each).
- In all precincts, for the remaining 9 contests,
  345 ballots shall vote for ballot position 1 in every contest
  345 ballots shall vote for ballot position 2 in every contest
  1 ballot shall vote for ballot position 3 in contest 3 and undervote the rest
  1 ballot shall vote for ballot position 4 in contest 4 and undervote the rest
  1 ballot shall vote for ballot position 5 in contest 5 and undervote the rest
  1 ballot shall vote for ballot position 6 in contest 6 and undervote the rest
  1 ballot shall vote for ballot position 7 in contest 7 and undervote the rest
  1 ballot shall vote for ballot position 8 in contest 8 and undervote the rest
  1 ballot shall vote for ballot position 9 in contest 9 and undervote the rest
  1 ballot shall vote for ballot position 10 in contest 10 and undervote the rest

In precincts 1 through 50, all remaining ballots shall vote for ballot position 1 in the first contest and undervote the rest.

In precincts 51 through 100, all remaining ballots shall vote for ballot position 2 in the first contest and undervote the rest.

4.2.3.10.4 Discussion

Write-ins, etc. need a typical case test that combines all typical profiles, if there is a most common combination.

4.2.3.11 Capacity tests

Following subsections are organized by compliance profiles. Functional tests are applicable only if the Implementation Statement asserts conformance to the profile indicated in the subsection name.

Special instructions for Optical Scan and Punchcard: For systems claiming conformance to the *Optical Scan* or *Punchcard* profiles, all applicable capacity tests shall be executed at the maximum speed or rate at which the tabulating equipment is documented to function reliably.

4.2.3.11.1 All systems

4.2.3.11.1.1 **Test case name:** 1-of-M Contest/Ballot Capacity

Ballot form: L_R 1-of-M contests where M = L_C.
The contests shall be described as follows (substituting numbers from 1 to L_R for *r*):

> This is Contest *r* in the 1-of-M Contest/Ballot Capacity Test. Vote for at most one.

The ballot positions in each contest shall be of the following form (substituting numbers from 1 to L_C for *c*):

> Contest *r* Ballot Position *c*

There are no write-in ballot positions in this ballot form.
Reporting contexts: Single context.
Scenario:
A total of L_B ballots shall be cast.
$\min\left(L_T, \max\left(\left\lceil \frac{L_B}{L_C} \right\rceil - c, 0\right)\right)$ ballots shall vote for Ballot Position *c* in every contest ($c > 0$).

Any ballots left over shall be blank (undervotes).

4.2.3.11.1.2 **Test case name:** Ballot Form Capacity

Assumptions: $L_T \geq \min(L_B, L_F) - L_C + 1$
Ballot forms:

L_F ballot forms shall be constructed. These forms shall share the same set of contests. (They shall be identical except for their form identifications.)
There shall be L_R 1-of-M contests where M = L_C.
The contests shall be described as follows (substituting numbers from 1 to L_R for *r*):
This is Contest *r* in the Ballot Form Capacity Test. Vote for at most one.
The ballot positions in each contest shall be of the following form (substituting numbers from 1 to L_C for *c*):
Contest *r* Ballot Position *c*
There are no write-in ballot positions in any of the ballot forms.
Reporting contexts: Single context.
Scenario:
$\min(L_B, L_F)$ ballots shall be cast.

The *n*th ballot shall use ballot form *n* and shall vote for ballot position $\min(n, L_C)$ in every contest.

4.2.3.11.1.3 **Test case name:** Vote Register Capacity

Ballot form: 1 1-of-M contest where M = 1.
The contest shall be described as follows:
This is the only contest in the Vote Register Capacity Test. There is only one candidate on the ballot.
The only ballot position in the contest shall be the following:
Unopposed Candidate
Reporting contexts: Single context.
Scenario: A total of $\min(L_B, L_T)$ ballots shall be cast. All shall vote for Unopposed Candidate.

4.2.3.11.1.4 **Test case name:** Undervote Register Capacity

Ballot form: 1 1-of-M contest where M = 1.
The contest shall be described as follows:
This is the only contest in the Undervote Register Capacity Test. There is only one candidate on the ballot.
The only ballot position in the contest shall be the following:
Unopposed Candidate
Reporting contexts: Single context.
Scenario: A total of $\min(L_B, L_T)$ ballots shall be cast. All shall be blank.

4.2.3.11.1.5 **Test case name:** 1-of-M Multi Capacity

Assumptions: $L_R \geq L_F$
Ballot forms:
L_F ballot forms shall be constructed. The first contest on each form shall be unique to that form. It shall be described as follows (substituting numbers from 1 to L_F for *f*):

This contest appears only in Ballot Form *f*. Vote for at most one.

The ballot positions in these contests shall be of the following form (substituting numbers from 1 to L_C for *c*):

Form *f* Position *c*

Each ballot form shall contain L_F - L_R other contests that are shared by all ballot forms. These contests shall be described as follows (substituting numbers from 1 to L_F - L_R for *r*):
This is Shared Contest *r* in the 1-of-M Multi Capacity Test. Vote for at most one.
The ballot positions in each contest shall be of the following form (substituting numbers from 1 to L_C for *c*):

Contest *r* Ballot Position *c*

There are no write-in ballot positions in any of the ballot forms.
Reporting contexts: Single context.
Scenario:

– For each ballot form *f* , for *f* from 1 to L_F, $\left\lfloor \frac{L_B}{L_F} \right\rfloor$ ballots shall be cast. Let $n = \left\lfloor \frac{L_B}{L_F} \right\rfloor \times L_F$. (This may total fewer than L_B ballots.)

– In the contest that is unique to each ballot form, $\min\left(L_T, \max\left(\left\lceil \frac{\left\lfloor \frac{L_B}{L_F} \right\rfloor}{L_C} \right\rceil - c, 0 \right) \right)$ ballots shall vote for Ballot Position *c* (*c* > 0). Any ballots left over shall undervote the unique contest.

In all other (shared) contests, $\min\left(L_T, \max\left(\left\lceil \frac{n}{L_C} \right\rceil - c, 0 \right) \right)$ ballots shall vote for Ballot Position *c* in every contest (*c* > 0). Any ballots left over shall undervote all of the shared contests.

4.2.3.11.2 Optical Scan and Punchcard

4.2.3.11.2.1 **Test case name:** Overvote Register Capacity

Ballot form: 1 1-of-M contest where M = 2.
The contest shall be described as follows:

This is the only contest in the Overvote Register Capacity Test. There are two candidates on the ballot. Vote for one.

The ballot position in the contest shall be the following:

Ballot Position 1
Ballot Position 2

Reporting contexts: Single context.
Scenario: A total of $\min(L_B, L_T)$ ballots shall be cast. All shall vote for both ballot positions (overvote).

4.2.3.11.3 Write-ins

4.2.3.11.3.1 **Test case name:** 1-of-M Write-in Capacity 1

Ballot form: L_R 1-of-M contests where M = L_C.
The contests shall be described as follows (substituting numbers from 1 to L_R for *r*):

This is Contest *r* in the 1-of-M Write-in Capacity 1 Test. Vote for at most one.

The ballot positions from 1 to L_C-1 in each contest shall be of the following form (substituting numbers from 1 to L_C-1 for *c*):

Contest *r* Ballot Position *c*

The final ballot position in each contest shall be a write-in opportunity.
Reporting contexts: Single context.
Scenario: A total of $\min(L_B, L_T)$ ballots shall be cast. All shall write in "Write-in Candidate *r*" in every contest, substituting contest numbers 1 to L_R for *r*.

4.2.3.11.3.2 **Test case name:** 1-of-M Write-in Capacity 2

Ballot form: L_R 1-of-M contests where M = L_C.
The contests shall be described as follows (substituting numbers from 1 to L_R for *r*):

This is Contest *r* in the 1-of-M Write-in Capacity 2 Test. There are no candidates on the ballot. Write in at most one.

The only ballot position in each contest shall be a write-in opportunity.
Reporting contexts: Single context.
Scenario: A total of $\min(L_B, L_T, L_W)$ ballots shall be cast. All shall write in "Write-in Candidate *n*" in every contest, substituting ballot numbers 1 to $\min(L_B, L_T, L_W)$ for *n* (each ballot shall vote for a different write-in candidate).

4.2.3.11.4 Straight party voting

4.2.3.11.4.1 **Test case name:** 1-of-M Straight Party Capacity

Ballot form:
The first contest shall be described as follows:

STRAIGHT PARTY. If you desire to vote a straight party ticket for all offices, vote for at most one party here. Votes for individual candidates in subsequent contests will override the straight party vote in those contests only.

The only ballot position shall be the following:

Whig

There shall be L_R-1 1-of-M contests where M = L_C.
The contests shall be described as follows (substituting numbers from 1 to L_R-1 for *r*):

This is Contest *r* in the 1-of-M Straight Party Capacity Test. Vote for at most one.

The first ballot position in each contest shall be of the following form:

Contest *r* Whig Candidate (Whig)

The remaining ballot positions in each contest shall be of the following form (substituting numbers from 2 to L_C for *c*):

Contest *r* Ballot Position *c*

There are no write-in ballot positions in this ballot form.
Reporting contexts: Single context.
Scenario: A total of $\min(L_B, L_T)$ ballots shall be cast. Each one shall vote straight party Whig in the first contest and skip the remaining contests (allowing the straight party vote to be effective in every contest).

4.2.3.11.5 N of M voting

4.2.3.11.5.1 **Test case name:** N-of-M Capacity

Ballot form: L_R N-of-M contests where N = M = L_C.
The contests shall be described as follows (substituting numbers from 1 to L_R for *r*):

This is Contest *r* in the N-of-M Capacity Test. Vote for at most L_C.

The ballot positions in each contest shall be of the following form (substituting numbers from 1 to L_C for *c*):

Contest *r* Ballot Position *c*

There are no write-in ballot positions in this ballot form.
Reporting contexts: Single context.
Scenario: A total of $\min(L_B, L_T)$ ballots shall be cast. All shall vote for every candidate in every contest.

4.2.3.11.6 N of M voting + Write-ins

4.2.3.11.6.1 **Test case name:** N-of-M Write-ins Capacity 1

Ballot form: L_R N-of-M contests where N = $\left\lfloor \frac{L_C}{2} \right\rfloor$ and M = L_C.

The contests shall be described as follows (substituting numbers from 1 to L_R for *r*):

This is Contest *r* in the N-of-M Write-ins Capacity 1 Test. Vote for at most N.

The first M-N ballot positions in each contest shall be of the following form (substituting numbers from 1 to M-N for *c*):

Contest *r* Ballot Position *c*

The final N ballot positions shall be write-in opportunities.

Reporting contexts: Single context.

Scenario: A total of $\min(L_B, L_T)$ ballots shall be cast. $\left\lfloor \frac{\min(L_B, L_T)}{2} \right\rfloor$ ballots shall vote for the first N candidates in each contest. The remaining ballots shall write in N candidates of the form "Contest *r* Write-in *n*," for *n* from 1 to N, in each contest.

4.2.3.11.6.2 **Test case name:** N-of-M Write-ins Capacity 2

Ballot form: L_R N-of-M contests where N = M = L_C.

The contests shall be described as follows (substituting numbers from 1 to L_R for *r*):

This is Contest *r* in the N-of-M Write-ins Capacity 2 Test. There are no candidates on the ballot. Write in at most L_C choices.

All L_C ballot positions in each contest shall be write-in opportunities.

Reporting contexts: Single context.

Scenario: A total of $\min(L_B, L_T)$ ballots shall be cast. In every contest, every ballot shall write in L_C choices of the form "Contest *r* Write-in *c*," substituting numbers 1 to L_C for *c*.

4.2.3.11.7 Cumulative voting

4.2.3.11.7.1 **Test case name:** Cumulative Voting Capacity

Ballot form: L_R cumulative voting contests where M = N(*r*) = L_C.

The contests shall be described as follows (substituting L_C and numbers from 1 to L_R for *r*):

This is Contest *r* in the Cumulative Voting Capacity Test. Cast at most L_C votes. You may cast multiple votes for the same candidate.

The ballot positions in each contest shall be of the following form (substituting numbers from 1 to L_C for *c*):

Contest *r* Ballot Position *c*

There are no write-in ballot positions in this ballot form.

Reporting contexts: Single context.

Scenario: A total of $\min\left(L_B, \left\lceil \frac{L_T}{L_C} \right\rceil\right)$ ballots shall be cast.

$\min\left(L_B, \left\lfloor \frac{L_T}{L_C} \right\rfloor\right)$ ballots shall cast L_C votes for the first ballot position in every contest.

Any ballot left over shall cast $L_T - \left\lfloor \frac{L_T}{L_C} \right\rfloor \times L_C$ votes for the first ballot position in every contest and undervote the rest.

4.2.3.11.8 Provisional / challenged ballots

4.2.3.11.8.1 **Test case name:** Provisional Ballot Capacity

Ballot form: L_R 1-of-M contests where M = L_C.
The contests shall be described as follows (substituting numbers from 1 to L_R for *r*):

> This is Contest *r* in the Provisional Ballot Capacity Test. Vote for at most one.

The ballot positions in each contest shall be of the following form (substituting numbers from 1 to L_C for *c*):

> Contest *r* Ballot Position *c*

There are no write-in ballot positions in this ballot form.
Reporting contexts: Single context.
Scenario: A total of $\min(L_B, L_V)$ provisional ballots shall be cast and accepted for counting.

$\min\left(L_T, \max\left(\left\lceil \frac{\min(L_B, L_V)}{L_C} \right\rceil - c, 0\right)\right)$ ballots shall vote for Ballot Position *c* in every contest ($c > 0$).
Any ballots left over shall be blank (undervotes).
Additional pass criteria: The number of provisional / challenged ballots cast, read and counted shall be reported in compliance with FR4.3.5.1, FR4.3.6.3, and FR4.3.7.2.

4.2.3.11.9 Discussion

In all systems there is the possibility of a bottleneck in transmitting precinct results to central. Unfortunately there is no way to test this without having a large number of machines to work with – an unreasonable expectation for a system that is not even qualified yet.

Many more capacity tests could be written for different combinations of profiles. There should be at least one torture test for the most common combination of profiles, if there is one.

4.2.3.12 Error case tests

While most tests verify that the system does things that it is required to do, error case tests verify that the system does not do things that it is required not to do. As with all tests, passing an error case test does not conclusively show that the system is conforming, but failing an error case test conclusively shows that the system is non-conforming.

4.2.3.12.1 All systems

4.2.3.12.1.1 Test case name: Vote Register Overflow

Assumptions: $L_B > L_T$
Ballot form: 1 1-of-M contest where M = 1.
The contest shall be described as follows:
This is the only contest in the Vote Register Overflow Test. There is only one candidate on the ballot.
The only ballot position in the contest shall be the following:
Unopposed Candidate
Reporting contexts: Single context.
Scenario: The VSTL shall attempt to cast L_T+1 ballots, all voting for Unopposed Candidate.
Additional pass criteria: A DRE system shall not enable the L_T+1th ballot. The tabulator in an Optical Scan or Punchcard system shall not accept the L_T+1th ballot. (Revised coding standards, v1s4, 4.2.2, Software Integrity)
An audit log record shall exist for the counter reaching capacity event.
Discussion: Overflow of the ballot counter is also implicitly tested by all of these register overflow tests.

4.2.3.12.1.2 Test case name: Undervote Register Overflow

Assumptions: $L_B > L_T$
Ballot form: 1 1-of-M contest where M = 1.
The contest shall be described as follows:
This is the only contest in the Undervote Register Overflow Test. There is only one candidate on the ballot.
The only ballot position in the contest shall be the following:
Unopposed Candidate
Reporting contexts: Single context.
Scenario: The VSTL shall attempt to cast L_T+1 ballots, all of them blank.
Additional pass criteria: A DRE system shall not enable the L_T+1th ballot. The tabulator in an Optical Scan or Punchcard system shall not accept the L_T+1th ballot. (Revised coding standards, v1s4, 4.2.2, Software Integrity)
An audit log record shall exist for the counter reaching capacity event.

4.2.3.12.2 Optical Scan and Punchcard

4.2.3.12.2.1 **Test case name:** Overvote Register Overflow

Assumptions: $L_B > L_T$
Ballot form: 1 1-of-M contest where M = 2.
The contest shall be described as follows:
This is the only contest in the Overvote Register Overflow Test. There are two candidates on the ballot. Vote for one.
The ballot position in the contest shall be the following:
Ballot Position 1
Ballot Position 2
Reporting contexts: Single context.
Scenario: The VSTL shall attempt to cast L_T+1 ballots. All shall vote for both ballot positions (overvote).
Additional pass criteria: The tabulator shall not accept the L_T+1th ballot. (Revised coding standards, v1s4, 4.2.2, Software Integrity)
An audit log record shall exist for the counter reaching capacity event.

4.2.3.12.3 DRE

4.2.3.12.3.1 **Test case name:** DRE Overvoting

Ballot form: 1 1-of-M contest where M = 3.
The contest shall be described as follows:
This is the only contest in the DRE Overvoting Test. There are three candidates on the ballot. Vote for at most one.
The ballot positions shall be the following:
Ballot Position 1
Ballot Position 2
Ballot Position 3
Reporting contexts: Single context.
Scenario: The VSTL shall attempt to cast one ballot that votes for both Ballot Position 2 and Ballot Position 3.
Additional pass criteria: The DRE shall prevent the VSTL from voting for more than one ballot position. (FR4.1.3.5)

4.2.3.12.4 DRE + Write-ins

4.2.312.4.1 **Test case name:** DRE Write-ins Overvoting

Ballot form: 1 1-of-M contest where M = 2.
The contest shall be described as follows:
This is the only contest in the DRE Write-ins Overvoting Test. There is one candidate on the ballot. Vote for at most one.

The first ballot position shall be the following:

Ballot Position 1

The second ballot position in the contest shall be a write-in opportunity.

Reporting contexts: Single context.

Scenario: The VSTL shall attempt to cast one ballot that both votes for Ballot Position 1 and writes in "Write-in candidate."

Additional pass criteria: The DRE shall prevent the VSTL from voting for more than one ballot position. (FR4.1.3.5)

4.2.3.12.5 N of M voting

4.2.3.12.5.1 **Test case name:** N-of-M Vote Register Overflow

Assumptions: $L_C >= 10$

$$L_B > \left\lfloor \frac{L_T}{10} \right\rfloor$$

Ballot form: 1 N-of-M contest where N = M = 10.

The contest shall be described as follows (substituting numbers from 1 to L_R for *r*):

This is the only contest in the N-of-M Vote Register Overflow Test. Vote for at most 10.

The ballot positions in each contest shall be of the following form (substituting numbers from 1 to 10 for *c*):

Ballot Position *c*

There are no write-in ballot positions in this ballot form.

Reporting contexts: Single context.

Scenario: The VSTL shall attempt to cast $\left\lfloor \frac{L_T}{10} \right\rfloor + 1$ ballots, all of which vote for all 10 ballot positions.

Additional pass criteria: A DRE system shall not enable the $\left\lfloor \frac{L_T}{10} \right\rfloor + 1$ th ballot.

The tabulator in a Optical Scan or Punchcard system shall not accept the $\left\lfloor \frac{L_T}{10} \right\rfloor + 1$ th ballot. (Revised coding standards, v1s4, 4.2.2, Software Integrity)

An audit log record shall exist for the counter reaching capacity event.

[Discussion: A likely fault is for the system to only check that the count is less than L_T before enabling a ballot (should be less than L_T-9). This fault is masked in this test if L_T is a multiple of 10. We could test with other values, but it is possible to mask the fault in all tests by making L_T a product of those values.]

[Discussion: Mechanical / lever systems appear to be out of scope of the VVSG. But if not – what is supposed to happen in mechanical / lever systems when a register reaches its limit?]

Other potential error case tests, not yet written:

-Remote data delivery with intentional disruption of transmission (implementation-dependent).

-Exception handling (requires creating an exception somehow – for testability, suggest adding a requirement for the capability to generate test exceptions. Could be of use in *in situ* L&A testing.)

4.2.3.13 Implementation-dependent structural tests

The VSTL shall review the vendor's program analysis, documentation, and, if available, module test case design. The VSTL shall evaluate the test cases for each module, with respect to flow control parameters and data on both entry and exit. All discrepancies between the Software Specifications and the test case design shall be corrected by the vendor prior to initiation of the qualification test.

If the vendor's module test case design does not provide conclusive coverage of all program paths, then the VSTL shall perform an independent analysis to assess the frequency and consequence of error of the untested paths. The VSTL shall define and execute additional module test cases as required to provide coverage of all modules containing untested paths with potential for untrapped errors. The VSTL shall define pass criteria for implementation-dependent structural tests using the VVSG and the vendor-supplied system documentation to determine acceptable ranges of performance.

The VSTL shall report the implementation-dependent structural tests performed and the test verdicts. No system shall be qualified if any implementation-dependent structural tests are assigned the verdict Fail using the VSTL's defined pass criteria.

[Discussion: This text is retained from the 2002 VSS II.A.4.3.3, "Software Module Test Case Design and Data," with minor changes. As time permits, this section should be rewritten to enhance repeatability and reproducibility of the testing.]

4.2.3.14 Implementation-dependent functional tests

The VSTL shall review the vendor's functional test case designs. The VSTL shall prepare a detailed matrix of system functions and the test cases that exercise them. The VSTL shall also prepare a test procedure describing all test ballots, operator procedures, and the data content of output reports. Abnormal input data and operator actions shall be defined. Test cases shall also be designed to verify that the system is able to handle and recover from these abnormal conditions.

The vendor's test case design may be evaluated by any standard or special method appropriate.

In the event that the vendor's functional test data are insufficient, the VSTL shall define and execute additional functional tests. The VSTL shall define pass criteria for implementation-dependent functional tests using the VVSG and the vendor-supplied system documentation to determine acceptable ranges of performance.

Depending upon the design and intended use of the voting system, all or part of the functions listed below shall be tested.

- Ballot preparation subsystem;
- Test operations performed prior to, during, and after processing of ballots, including:
    - Logic tests to verify interpretation of ballot styles, and recognition of precincts to be processed;
    - Accuracy tests to verify ballot reading accuracy;
    - Status tests to verify equipment statement and memory contents;
    - Report generation to produce test output data; and
    - Report generation to produce audit data records;
- Procedures applicable to equipment used in the polling place for:
    - Opening the polling place and enabling the acceptance of ballots;
    - Maintaining a count of processed ballots;
    - Monitoring equipment status;
    - Verifying equipment response to operator input commands;
    - Generating real-time audit messages;
    - Closing the polling place and disabling the acceptance of ballots;
    - Generating election data reports;
    - Transfer of ballot counting equipment, or a detachable memory module, to a central counting location; and
    - Electronic transmission of election data to a central counting location; and
- Procedures applicable to equipment used in a central counting place:
    - Initiating the processing of a ballot deck, programmable memory device, or other applicable media for one or more precincts;
    - Monitoring equipment status;
    - Verifying equipment response to operator input commands;
    - Verifying interaction with peripheral equipment, or other data processing systems;
    - Generating real-time audit messages;
    - Generating precinct-level election data reports;
    - Generating summary election data reports;
    - Transfer of a detachable memory module to other processing equipment;
    - Electronic transmission of data to other processing equipment; and
    - Producing output data for interrogation by external display devices.

The VSTL shall report the implementation-dependent functional tests performed and the test verdicts. No system shall be qualified if any implementation-dependent functional tests are assigned the verdict Fail using the VSTL's defined pass criteria.

[Discussion: This text is retained from the 2002 VSS II.A.4.3.4, "Software Functional Test Case Design," with minor changes. As time permits, this section should be rewritten to enhance repeatability and reproducibility of the testing.]

4.2.4 Analysis of the 2002 VSS Requirements

This document presents the results of an analysis of the requirements contained in the "Voting Systems Performance and Test Standards" ("VSS"), released by the Federal Election Commission in 2002. The analysis was performed by the NIST voting team in support of the Election Assistance Commission's Technical Development Guideline Committee (TGDC) resolution 25-05. The objectives of the analysis were to:

- thoroughly review the VSS to determine which parts of the VSS can be extracted or recast into a new voting system standard, called the Voluntary Voting System Guidelines (VVSG)
- identify which parts of the VSS require substantial changes or replacement
- identify the VSS requirements that are and are not testable
- help identify omissions in the VSS that have resulted from changes in technology or other reasons.

This analysis will be a starting point for the development of the VVSG version 2.

4.2.4.1 Approach

The VSS comprises two volumes: "Volume I, Voting System Performance Standards" and "Volume II, Voting System Test Standards." The normative text in each volume (sections 2 through 9 of Volume I and sections 2 through 7 of Volume II) was carefully examined and the individual requirements identified. Each requirement was then evaluated as to its disposition in the VVSG: whether it should be retained as is, revised, deleted, or moved. This information was then placed in Table 1, corresponding to Volume I, or Table 2, corresponding to Volume II, as appropriate. Each requirement in the tables was assigned to the TGDC subgroup that was considered to be the subgroup most responsible for that requirement. Each requirement was also associated with the Organizing Principle that was considered to be the one most closely supported by that requirement. Finally, any observations and open questions were recorded.

4.2.4.2 Summary of the Analysis

Much of the VSS is valid and can be retained in the VVSG, either as-is or rewritten to be more precise. Most of the current functional and hardware requirements can stay largely intact, but some need to be made more precise and testable, and others need to be updated to correspond to current technology. The principal areas that require more major changes are the VSS

requirements dealing with accessibility and usability, security, software design and coding standards, configuration management, and quality assurance. More specifically:

- The sections in the VSS on accessibility (Vol. I, section 2.2.7) and human engineering (Vol. I, section 3.4.9), and the guidelines on usability (Vol. I, Appendix C), need to be replaced in the VVSG. The text in the VSS is little more than a placeholder.

- Requirements dealing with voter verifiable paper audit trails, wireless communications, software distribution, and direct and indirect vote verification need to be added. The VSS sections dealing with protection against external threats need to be rewritten to better address electronic voting systems.

- The VSS sections on software design and coding standards (primarily Vol. I, section 4.2 and Vol. II, section 5.4) are obsolete and need to be replaced, per resolution 29-05.

- The VSS sections on configuration management and quality assurance (primarily Vol. I, sections 7 and 8, and Vol. II, section 7) need to be replaced, per resolution 30-05.

4.2.4.3 The Analysis Tables

The Analysis Tables (Appendix C), namely Table 1 corresponding to VSS Volume I, and Table 2, corresponding to VSS Volume II, are presented in the Appendix.

4.2.4.3.1 How to Read the Tables

Each table consists of 6 columns. In a given row,

- Column 1 is the number of that row.

- Column 2 ("**G**") is a letter representing the TGDC subgroup with primary responsibility for the requirement contained in that row. Some requirements are assigned to more than one subgroup. The letters are:
  - **C** Core Requirements and Testing
  - **H** Human Factors and Privacy
  - **S** Security and Transparency

- Column 3 ("**Section/Requirement**") contains either a section heading in the VSS, or a requirement identified in and extracted from that section. The requirements are essentially those statements in the VSS specifying that some entity "shall" do something.

- Column 4 ("**T**") is a letter representing the type of disposition proposed for the given requirement. The letters are:
  - **E** The requirement is satisfactory as currently written in the VSS, and can be extracted and retained as-is. There is no need to rewrite it.

  - **R** The requirement is not precise, clear, or testable, or contains a typo, bad reference, or other mistake. It needs to be rewritten.

  - **M** In Table 1: The requirement is not a performance requirement, but rather a testing requirement. It should be moved to Volume II of the VSS.
    In Table 2: The requirement is not a testing requirement, but rather a performance requirement. It should be moved to Volume I of the VSS.
  - **D** The requirement is obsolete, redundant, or otherwise unnecessary. It should be deleted.

  Disposition is not proposed for requirements in those VSS sections that will be entirely rewritten. When this situation occurs, it is noted in column 6.

- Column 5 ("**P**") is the number of the organizing principle (reference) that the given requirement supports. Some requirements support more than one organizing principle.

- Column 6 ("**Comments**") contains questions, answers to previously submitted questions, and other observations about the given requirement. For many requirements, there is an indication as to whether or not the requirement is testable. Questions are highlighted in **bold gray.**

4.2.4.4 What We Found

Table 1 (Appendix C)

- There are **541** identified requirements for which disposition is proposed.

- The assignment of identified requirements to the TGDC subgroups is as follows:
  - **394** requirements assigned to the CRT subgroup
  - **72** requirements assigned to the HFP subgroup
  - **126** requirements assigned to the STR subgroup.

  (Some requirements are assigned to more than one subgroup.)

- The disposition of requirements is as follows:
  - **310** requirements proposed to be extracted
  - **169** requirements proposed to be rewritten
  - **32** requirements proposed to be moved to Volume II
  - **30** requirements proposed to be deleted.

- The association of requirements to organizing principles is as follows:
  - **8** requirements associated with Principle 1
  - **32** requirements associated with Principle 2
  - **71** requirements associated with Principle 3
  - **57** requirements associated with Principle 4
  - **44** requirements associated with Principle 5
  - **378** requirements associated with Principle 6
  - **112** requirements associated with Principle 7

– **41** requirements associated with Principle 8.

(Some requirements are associated with more than one organizing principle.)

Table 2 (Appendix C)

- There are **344** identified requirements for which disposition is proposed.

- The assignment of identified requirements to the TGDC subgroups is as follows:
  – **299** requirements assigned to the CRT subgroup
  – **31** requirements assigned to the HFP subgroup
  – **67** requirements assigned to the STR subgroup.

  (Some requirements are assigned to more than one subgroup.)

- The disposition of requirements is as follows:
  – **282** requirements proposed to be extracted
  – **59** requirements proposed to be rewritten
  – **0** requirements proposed to be moved to Volume I
  – **3** requirements proposed to be deleted.

- The association of requirements to organizing principles is as follows:
  – **6** requirements associated with Principle 1
  – **3** requirements associated with Principle 2
  – **2** requirements associated with Principle 3
  – **10** requirements associated with Principle 4
  – **0** requirements associated with Principle 5
  – **323** requirements associated with Principle 6
  – **51** requirements associated with Principle 7
  – **1** requirement associated with Principle 8.

  (Some requirements are associated with more than one organizing principle.)

4.3 Requirements for Principle 2.8

Security Overview

This section addresses four new, specific aspects of voting systems security. These new items are:

1. Definitions for Independent Verification Voting Systems: definition of voting systems that produce multiple records of votes. A future version of the VVSG will require that voting systems produce multiple records of ballots or receipts for auditing purposes.

2. Security Requirements for Voter Verified Paper Audit Trails: requirements for voter verified paper audit trails, if a State chooses to require them.

3. Use of Wireless Networking in Voting Systems: how wireless networks and the data sent across wireless networks should be secured.

4. Security Requirements for Software Distribution and Setup Validation of Voting System: requirements for the secure distribution of voting systems software and ballot information for verifying that voting systems are operating with the correct software and software configuration.

The remainder of this section is an informative section with discussion of independent verification systems followed by definitions of the types of independent verification systems, which will be used as the basis for future requirements. The definitions are preliminary and will be evolving with further research.

4.3.1 Independent Verification Systems (Informative)

The primary objective for using electronic voting systems is the production of voting records that are highly precise, highly reliable, and easily counted - in essence, an accurate representation of ballot choices whose handling requirements are reasonable. To meet these objectives, there are many factors to consider in an electronic voting system's design, including:

- the environment provided for voting, including the voting site and various environmental factors,
- the ease with which voters can use the voting system, i.e., its usability,
- the robustness and reliability of the voting equipment, and
- the capability of the records to be used in audits.

Independent Verification systems have as their primary objective the production of ballot records that are capable of being used in audits in which their correctness can be audited to very high levels of precision. The primary security issues addressed by independent verification systems are:

- whether electronic voting systems are accurately recording ballot choices, and
- whether the ballot record contents can be audited precisely post-election.

The threats addressed by independent verification systems are those that could cause a voting system to inaccurately record the voter's intent or cause a voting system's records to become damaged, i.e., inserted, deleted, or changed. These threats could occur via any number of means including accidental damage or various forms of fraud. The threats are addressed mainly by providing, in the voting system design, the capability for ballot record audits to detect precisely whether specific records are correct as recorded or damaged, missing, or fraudulent.

4.3.1.1 Problems in Auditing Single Record Voting Systems

The auditing paradigm in financial transactions, e.g., transactions in which a merchant retains a copy of the transaction and the purchaser retains a receipt that can be reviewed for accuracy, does not apply for voting systems. This poses a complication for election officials and voters when

seeking the same high degrees of assurance that ballots cast on electronic voting systems are being recorded and counted correctly.

Electronic voting systems that produce a sole record of cast ballots are inherently limited in their capability for accurate audits - as would a financial system that produced only one record of its transactions[11]. When there is only one record, the assurance that the cast ballots are being correctly recorded by the voting system is limited to other means such as:

- confidence in how well the voting system was inspected and tested,
- logic and accuracy tests performed pre-election,
- parallel testing of voting equipment on election day,
- inspection of the voting system's event log for anomalous behavior,
- comparison of election results with post-election polls, and
- comparison of election results with expected voter behavior.

It is highly desirable that electronic voting systems be designed such that they already include, as a fundamental part of their design, the mechanisms to provide highly accurate and reliable auditing of ballot contents.

4.3.1.2 Independent Verification Systems: Improved Accuracy in Audits

Independent Verification is the top-level categorization for electronic voting systems that produce multiple records of ballot choices whose contents are capable of being audited to high levels of precision. For this to happen, the records must be produced, verified by the voter, and subsequently handled according to the following protocol:

(a) At least two records of the voter's choices are produced and one of the records is then stored such that it cannot be modified by the voting system, e.g. the voting system creates a record of the voter's choices and then copies it to some write-once media.

(b) The voter must verify that both records are correct, e.g., verify his or her choices on the voting system's display and also verify the second record of choices stored on the write-once media.

(c) The verification processes for the two verifications must be independent of each other and (a) at least one of the records must be verified directly by the voter, or (b) it is acceptable for the voter to indirectly verify both records if they are stored on different systems produced by different vendors.

(d) The content of the two records can be checked later for consistency through the use of identifiers that allow the records to be linked.

[11] Electronic voting systems that create and store copies of their electronic records or that print a copy of their electronic records in effect store just one record of cast ballots because the additional records are clones of the first record. The additional records cannot be used to audit the accuracy of the first record.

An assumption is made that at least one set of the records is usable in an efficient counting process, such as by using an electronic voting system, and the other set of records is usable in an efficient process of verifying its agreement with the first set of records. The other set records would preferentially be different in form from the first set of records and have some resistance to accidental or deliberate damage.

Given these conditions above, the multiple records are said to be *distinct* and *independently verifiable*, that is, both records are not under the control of the same processes. As a result of this independence, one record can be used to audit or check up on the accuracy of the other record. Because the storage of the records is separate, an attacker who can compromise one of these records still will face a difficult task in compromising the other.

A simple example of an independent verification system is an electronic voting station that records a voter's choices and then writes them to a token. If the voter removes the token and inserts it into a separate system that makes an electronic copy of the token and displays it to the voter, the voter can then verify that the first station has recorded the ballot correctly and the second station has copied and stored the ballot correctly. This example satisfies the four conditions necessary for handling multiple records in independent verification systems, as follows:

- Condition (a) is satisfied because two records are created and the record stored on the token cannot be modified by the same system used to create the electronic copy.

- Condition (b) is satisfied because the voter verifies at the second station that the record stored on the token is accurate and verifies at the second station that the copy of the token's record made by the second station is correct.

- Condition (c) is satisfied because the voter is able to directly verify that the record stored on the token is accurate -- the verification of the second record is indirect, because the same voting system that created the separate record is being used to verify it.

- Condition (d) satisfied because the records are created so that the record on the token can identify its copy stored by the voting system (this wasn't included in the example but is assumed to happen).

There are many types of independent verification systems. This example is a split process system, as described in Section 0.

4.3.1.3 Example Independent Verification Systems

The following sections contain informative overviews of several types of independent verification systems, some of which have not been implemented yet. Thus their inclusion in this document is intended to help clarify approaches to independent verification systems. The systems discussed are:

- voting systems with a split process architecture,
- end-to-end voting systems that include cryptographic audit schemes,
- witness voting systems that take a picture of or otherwise capture an indirect verification of ballot choices,
- direct independent verification, including some types of voting systems that produce an optically scanned ballot or that produce a voter-verified paper audit trail (VVPAT).

4.3.1.3.1 The Split Process Architecture for Independent Verification Systems

A voting machine in this scheme consists of vote capture and verification stations that are kept separate, i.e., two physical devices. A voter inserts an object called a token into the capture station to make ballot selections, and then takes the token object to the verification station to review and store his or her votes. The token object could be paper or some write-once read-only media. Two records of the vote are created: one on the token object and one by the verification station. Either could be used in the final count.[12]

Any split process voting system, the interaction between the voter and the split process is operates as follows:

1. A voter is given a token object that has been initialized to be blank.

2. Supporting information is written to the token object including the ballot and identification information about the election and precinct.

3. The voter inserts the token object into a capture station such as a DRE, which reads the ballot information from the token and then displays the ballot on an input device such as a touch screen. The voter then makes his or her ballot choices and then causes a record of the vote to be recorded on the token object.

4. The voter then takes the token object to a separate verification station, which reads the recorded votes from the token object, makes an electronic copy, and displays it to the voter.

5. The voter verifies that the information is correct and then deposits the token object into a container where it can be archived and used later for recounts or audits against the electronic records.

The electronic records recorded by the verification station typically would be counted in the election. One of the records should preferentially be different in form from the other record and

[12] The split process architecture is otherwise known as the frog protocol, which was first described in the Caltech – MIT report: *Voting: What Is, What Could Be*, as part of a modular voting architecture. The frog term, i.e., the token, was chosen specifically to convey no information about the physical form of the object used to carry vote information between two separate modules of the voting station. The report is available for download at http://www.vote.caltech.edu/.

have some resistance to accidental or deliberate damage so that it can remain useful for audits and recounts.

In theory, the physical separation of the ballot capture from the ballot verification may make analysis of the capture and verification devices easier or less costly. The rationale is that the user interface software on the capture station can be expected to be complex and difficult to verify for correctness. On the other hand, the verification station's software can be expected to be less complicated because it need only copy the contents of the token, display it to the voter, and then store the ballot choices.

The verification station's software can be considered to be the "trusted computing base" of the voting system, because it must be trusted in the verification process and then trusted to store the record for counting, i.e., cast the voter's ballot. Its software should be relatively small and thus easier to inspect and test.

In general, segregating functions by placing them on physically different systems is a standard computer security practice for making those functions easier to test for correctness and easier to manage securely.

4.3.1.3.2 End to End (Cryptographic) Independent Verification Systems

End to end voting systems use cryptographic techniques to store an encrypted copy of the voter's ballot choices and to give the voter the option to verify the correct recording and inclusion of his or her vote in the election totals. In this way, ballots can be audited and demonstrated to have been included in the final tally.

End to end systems in existence today generally operate as follows:

1. A voter uses a voting station such as a DRE to make ballot choices.

2. The DRE then issues a paper receipt to the voter that contains information that permits the voter to verify that the choices were recorded correctly. The information does not permit the voter, though, to reveal his or her choices.

3. The voter may have the option to check that his or her ballot choices were included in the final tally, e.g., by checking a web site of values that (should) match the information on the voter's paper receipt.

End to end systems are sometimes referred to as *receipt-based* systems. They may provide an assurance not only that the correct set of ballot choices was recorded, but also that those choices were included in the election count. Some analyses of auditing and cryptographic systems assert that very small numbers of self-audits are required to verify the correctness of an election.

4.3.1.3.3 Witness Independent Verification Systems

A witness voting system creates the second record of ballot choices by using a separate module to record or witness the voter's verification of the first record. The primary feature of a witness system that recommends itself is that the creation of the record does not require action by the voter. This may result in quicker voting times or voting systems that are simpler to use than some other schemes that involve multiple, direct verifications by the voter.

An example of a witness system is a DRE with a camera mounted above its screen. The camera takes pictures and saves them independently of the DRE. It would operate as follows:

1. A voter makes ballot choices at the DRE and then presses a button to record his or her vote.

2. The DRE records the ballot choices and uses them in the election count.

3. At the time the button is pressed, the camera takes a picture of the DRE's screen and saves the image (the voter is not included in the picture).

4. This collection of images constitutes a second ballot record that can be used in audits and recounts of the records recorded by the DRE.

As can be seen by this example, the voter's interactions are reduced to making ballot choices at the DRE and pressing a button to make the selections final. If the DRE were to have been compromised such that it secretly recorded the ballot choices incorrectly, the stored photographic images would reflect what the voter had seen and verified at the DRE's screen.

Because the voter cannot verify that the creation of the second record was performed accurately, a requirement of this type of system is that the creation process must be highly reliable and very resistant to accidental or deliberate damage. Also, the suitability of the records for manual or automated auditing must be considered in their selection.

4.3.1.3.4 Direct Independent Verification Systems

Direct independent verification systems produce a record for voter verification that the voter may verify directly with the voter's senses and which is then preserved for auditing or possibly counting. Some optical scan voting system schemes fit into this category (albeit loosely), as well as those systems with VVPAT (Voter Verified Paper Audit Trail) capability.

The type of optical scan voting systems schemes in this category are those in which two records are created: a paper and an electronic record. This system uses Optical Scan Recognition (OCR) to create an electronic record from the paper record after the paper record has been directly verified by the voter. The general operation of this system is:

1. A voter uses a marking device such as a DRE to mark a ballot and then presses a button to print the marked ballot onto a piece of paper.

2. The voter then directly reviews the paper to ensure its correctness, and if correct, places the paper record into a scanner (some procedure would need to be included to handle spoiled ballots).

3. The scanner converts the paper record into an electronic format. To reduce errors that may result from scanning the paper record, the paper records might contain a barcoded representation of the human readable portion of the ballot.

4. The paper record gets preserved in a ballot box.

The reason that the above scheme fits loosely into the independent verification category is because only one of the records was verified. One may assume that the scanning process is highly accurate and can be trusted to create the electronic record correctly; however it would be preferential for the voter to somehow verify that the record was, in fact, created correctly.

An electronic voting system with VVPAT (Voter Verified Paper Audit Trail) capability is similar to that of the optical scan above but consists typically of a DRE that both creates and records an electronic record, and printer to create a paper audit trail of the voter's choices. Like the optical scan system, it creates two distinct representations of the voters' ballot choices: an electronic record and a paper record.

Typically, a voter would use the voting system (called a DRE-VVPAT) as follows:

1. A voter makes ballot selections and then indicates that his or her selections are complete.

2. The VVPAT-DRE prints a paper record summary of the voter's ballot choices. An alternative approach to VVPAT involves printing the voter's ballot selections as they are made, e.g., a concurrent or contemporaneous record.

3. The voter inspects and directly verifies that the paper record matches the displayed electronic record (again, a procedure would need to be included to handle spoiled ballots).

4. The paper record gets preserved in a ballot box.

Both schemes described here produce paper records that are verified directly by sight. Voters with sight impairments require an accessible device for verification that can produce an audible representation of the paper record.

4.3.1.4 Issues in Handling Multiple Records Produced by Independent Verification Systems

There are several fundamental questions that need to be addressed when designing the structure and selecting the physical characteristics of independent verification voting systems records, including:

- how to tell if the records are authentic and not forged,
- how to tell if the integrity of the records has remained intact from the time they were recorded,
- the suitability of the records for various types of auditing, and
- how best to address problems if there are errors in the records.

Whenever an electronic voting system produces multiple records of votes, there is some possibility that one or more of the records may not match. Records can be lost, or deliberately or accidentally damaged, or stolen, or fabricated. Keeping the two records in correspondence with each other can be made more or less difficult depending on the technologies used for the records and the procedures used to handle the records.

As a consequence, it is important to structure the records so that errors and other anomalies can be readily detected during audits. There are a number of techniques that can be used, such as the following:

- associating unique identifiers with corresponding records, e.g., an individual paper record sharing a unique identifier with its corresponding electronic record,

- including an identification of the specific voting system that produced the records, such as a serial number identifier or by having the voting system digitally sign the records using public key cryptography,

- including other information about the election and the precinct or location where the records were created,

- creating checksums of the electronic records and having the voting system digitally sign the entire sets of records so that missing or inserted records can be detected, and

- structuring the records in open, publicly documented formats that can be readily analyzed on different computing platforms

The ease or relative difficulty with which some types of records must be handled is also a determining factor in the practical capability to conduct precise audits, given that some types of records are better suited to different types of auditing and different voting environments than others. The factors that make certain types of records more suitable than others could vary greatly depending upon many other criteria, both objective and subjective. For example, paper records may require manual handling by voters or poll workers and thus be more susceptible to damage or loss. At the same time, the extent to which the paper records must be handled will vary depending

on the type of voting system in use. Electronic records may by their nature be more suitable for automated audits; however electronic records are still subject to accidental or deliberate damage, loss, and theft.

It is not possible to discuss all factors and criteria that might make some records more suitable than others. Other procedures used in elections to help maintain the authenticity and integrity of records can also be affected by the suitability of the records, including procedures for comparing the count of cast ballots with the signatures of voters who cast the ballots, or procedures for maintaining accurate counts of how many ballots or cast on each voting system, or procedures for observing secure chains of custody of ballots. As stated previously, there may be subjective criteria for deciding which type of record is most suitable, e.g., a preference for paper despite its handling issues.

Lastly, the questions of what to do when problems occur and which records thus should be counted in the election can be difficult to answer. It can depend on which record is damaged, whether multiple records are damaged, and what the damage may indicate: ballot fraud, accidental damage, missing ballots, sabotage of the voting system, etc. Depending on how the records are damaged, it may require use of both records to reconstruct the complete record of voters' choices. Obviously, the more supporting evidence that is maintained in the structure of the record, the better equipped one is to make judgments as to which record to use.

4.3.2 Core Definitions for Independent Verification Systems (Informative)

This section contains a preliminary set of definitions for independent verification systems. These definitions are fundamental in nature and apply to all categories of independent verification systems. The remaining sections (following this section) contain definitions that are specific to those categories discussed in the preceding sections (split process, end to end, witness, and direct). The definitions will form the basis for future requirements for independent verification systems.

4.3.2.1 An independent verification voting system produces two distinct records of ballot choices via interactions with the voter whose equality of content can be audited to verify that the ballot choices were recorded accurately.

Responsible Entity: voting system vender
Process: Voting

Discussion: This is the fundamental core definition for independent verification systems. The records can be checked against one another to determine whether or not the voter's choices were being correctly recorded.

4.3.2.1.1 The voter verifies the content of each record and either (a) verifies at least one of the records directly or (b) verifies both records indirectly if the records are each under the control of independent processes.

Responsible Entity: voting system vender
Process: Voting

Discussion: A record can be verified directly by using senses, e.g., by sight, by ear. Indirect verification is when a technically and physically distinct module captures and makes a recording of the voter's verification of a record.

4.3.2.1.2 The creation, storage, and handling of the records are sufficiently separate such that the failure or compromise of one record does not cause the failure or compromise of another.

Responsible Entity: voting system vender
Process: Voting

Discussion: The records must be stored on different media and handled independently of each other, so that no one process could compromise all records. If an attack can alter one record, it should still be very difficult to alter the other record.

4.3.2.1.2.1 At least one record should be highly resistant to damage or alteration and should be capable of long-term storage.

Responsible Entity: voting system vender
Process: Voting

Discussion: Although not a requirement, at least one of the records should be difficult to alter or damage so that it could be used in case the counted records are damaged or lost.

4.3.2.1.3 The processes of verification for the multiple records do not all depend for their integrity on the same device, software module, or system, and are sufficiently separate such that the records each provide evidence of the voter's choices independently of the other records.

Responsible Entity: voting system vender
Process: Voting

Discussion: For example, the verification of an electronic record on a DRE is not sufficiently separate from the verification of an electronic record located on a token but performed on the same DRE as the verification for

the first record. Verification of a paper record by one's senses is sufficiently separate, in this case.

4.3.2.1.4 The records can be used in audits of one another, so that at least one set of records can be used in an efficient counting process, and another set of records can be used in an efficient process of verifying its substantial agreement with the first set of records.

Responsible Entity: voting system vender
Process: Voting

Discussion: For example, an electronic record can be used in an efficient counting process. A second paper record can be used to verify the accuracy of the electronic record; however its suitability for efficient counting is less clear. If a paper record can be used in an automated scan process, it may be more suitable.

4.3.2.1.5 The records include an identification of the voting site/precinct.

Responsible Entity: voting system vender
Process: Voting

Discussion: If the voting site and precinct are different, both should be included.

4.3.2.1.6 The records include information identifying whether the balloting is provisional, early, or on Election Day, and information that identifies the ballot style in use.

Responsible Entity: voting system vender
Process: Voting

4.3.2.1.7 The records include a voting session identifier that is generated when the voting station is placed in voting mode and that can be used to identify the records as being created during that voting session.

Responsible Entity: voting system vender
Process: Voting

Discussion: If there are several voting sessions on the same voting station on the same day, the voting session identifiers must be different. They should be generated from a random number generator.

4.3.2.1.8 The records include an identifier of the voting system that is unique to that style of voting systems.

Responsible Entity: Voting System
Process: Voting

Discussion: The identifier could be a serial number or other unique ID.

4.3.2.1.9 All cryptographic software in independent verification voting systems is in modules that have been approved by the U.S. Government's Crypto Module Validation Program (CMVP) as applicable.

Responsible Entity: voting system vender
Process: Voting

Discussion: The voting systems may use cryptographic software for a number of different purposes, including calculating checksums, encrypting records, authentication, generating random numbers, and for digital signatures. This software should be reviewed and approved by the Crypto Module Validation Program. There may be cryptographic voting schemes where the cryptographic algorithms used are necessarily different from any algorithms that have approved CMVP implementations, thus CMVP approved software should be used where feasible. The CMVP web site is http://csrc.nist.gov/cryptval.

4.3.3 Split Process Independent Verification Systems (Informative)

This section contains definitions specific to split process independent verification systems. The definitions build on and are in addition to the core definitions in Section 0. Split process systems consist of separate vote capture and verification stations that are kept separate, i.e., two physical devices. A voter inserts an object called a token into the capture station to make ballot selections, and then takes the token object to the verification station to review and store his or her votes. Two records of the vote are created: one on the token object and one by the verification station.

4.3.3.1 Capture and Verification Stations

4.3.3.1.1 The verification station is able to add information to the token object but cannot change prior recorded information

Responsible Entity: voting system vender
Process: Voting

Discussion: This will need to be evaluated by attempting to find a way to allow writing during penetration testing.

4.3.3.1.3 The capture and verification stations do not permit any communications between them except via the token object.

Responsible Entity: voting system vender
Process: Voting

4.3.3.1.3 The verification station log all rejected votes, including the votes' precise contents and an identifier of the token object.

Responsible Entity: voting system vender
Process: Voting

Discussion: The voter could reject and essentially spoil his or her ballot. If the verification station shows ballot choices that are different from what was entered at the capture station, this could be an indication of a serious problem.

4.3.3.1.4 The capture and verification stations could be purchased from different manufacturers and should use different operating systems.

Responsible Entity: voting system vender
Process: Voting

Discussion: The greater the diversity between the systems, the less likely they could be compromised by the same threats, e.g., software viruses, or by a single conspiracy.

4.3.3.2 Data Formats for Token Objects

4.3.3.2.1 The format for data written to the token object should be specified and available for use without permission or licensing fees.

Responsible Entity: voting system vender
Process: Voting

4.3.3.2.2 The verification station verifies the correctness of the data on the token object according to the specification of its format and provides an indication of any errors to the voter.

Responsible Entity: voting system vender
Process: Voting

Discussion: The verification station needs to verify, in essence, that the data written to the token object was formatted according to the rules of the format's specification and reject ill-formatted data. It also checks that the votes are consistent with the voting instructions, e.g., "vote for one, vote for two."

4.3.3.2.3 The record on the token object is digitally signed using a private key known only to the vote capture station and whose public key is distributed in an authenticated way to auditing systems.

Responsible Entity: voting system vender
Process: Voting

4.3.3.2.4 The record created by the verification station is digitally signed using a private key known only to the verification station and whose public key is distributed in an authenticated way to auditing systems.

Responsible Entity: voting system vender
Process: Voting

4.3.3.2.5 The capture station associates with each record of voter choices a unique identifier that is capable of being used to identify the record uniquely and to identify its corresponding record created by the verification station.

Responsible Entity: voting system vender
Process: Voting

Discussion: The identifier should serve the purpose of uniquely identify the record so as to identify duplicates and/or for cross-checking two record types

4.3.3.2.6 The records from the verification station are randomly shuffled in memory and when exported so that the order of the records cannot be used to identify any voter.

Responsible Entity: voting system vender
Process: Voting

Discussion:

4.3.3.2.7 Rejected token objects are stored separately from accepted memory devices for later auditing.

Responsible Entity: voting system vender
Process: Voting

4.3.3.3 Storage and Communications of Records

4.3.3.3.1 The verification station exports its records of voter choices accompanied by a digital signature on the entire set of electronic records and their associated digital signatures.

Responsible Entity: voting system vender
Process: Voting

Discussion: This is necessary to determine if records are missing or substituted.

4.3.3.3.2 The token objects are carried in a physically secure way, using chain-of-custody mechanisms to ensure their integrity.

Responsible Entity: voting system vender
Process: Voting

4.3.3.3.3 The records from each station are randomly shuffled, so that an attacker learning the contents of those records at any point in the voting can learn nothing about the order of votes cast.

Responsible Entity: voting system vender
Process: Voting

4.3.4 Witness Independent Verification Systems (informative)

This section contains preliminary definitions Witness independent verification systems. They are consistent with the definition of independent verification systems from this section and build on the core definitions from Section 0.

Witness independent verification systems are composed of two physically separate devices: the vote capture station that captures and stores records of voters' choices, and the witness device that captures voter verifications of the records at the vote recording station. Because there are two devices, a number of the definitions for split verification systems apply equally well to witness systems. Because the vote capture station is in essence a DRE (with or without VVPAT capability), a number of the definitions for VVPAT that are specific to DRE systems also apply to vote recording stations. A witness system fits somewhat loosely in the independent verification category because the voter performs only an indirect verification of ballot choices at the DRE and assumes that the witness device performs a second indirect verification. This assumption can be made only if the witness device is tested extensively for accuracy and reliability, and only if malfunctions in the device are made immediately obvious to voters and poll workers.

4.3.4.1 A witness device records only a voter's verification at a vote capture station and stores the record so that it can be used for audit and recounts as applicable.

Responsible Entity: voting system vender
Process: Voting

4.3.4.2 A witness device acts as a passive device that cannot perform any operation with respect to the capture station other than to capture the voter's ballot choices as the voter verifies them.

Responsible Entity: voting system vender
Process: Voting

Discussion: The witness device is synchronized with the voter verification of the ballot choices.

4.3.4.3 A witness device, if electrically connected to the capture station, is connected such that it can capture only the voter's verification of ballot choices.

Responsible Entity: voting system vender
Process: Voting

Discussion: For example, the witness device could be connected only to the display unit and not the capture device's memory or disk drive.

4.3.4.4 The capture station is not able to detect in its function whether a witness device is electrically connected or in operation.

Responsible Entity: voting system vender
Process: Voting

Discussion: If the witness device is connected to or attached electrically to the vote capture station, i.e., a DRE, the capture station is not able to determine or be aware in its function that a witness device is attached, other than its operating system would normally be able to determine that any device is attached to a hardware report under control of the operating system.

4.3.4.5 The witness device functions properly with most if not all electronic voting systems functioning as capture stations.

Responsible Entity: voting system vender
Process: Voting

Discussion: This is desirable but may possibly require some degree of openness in witness device specification so that voting system vendors could permit compatibility.

4.3.4.6 The witness device is not designed or built or manufactured by the same manufacturer of the capture station to which it is attached.

Responsible Entity: Testing Authorities
Process: Voting

4.3.4.7 Because voters must trust that the witness device records their verifications accurately, assessments of its software and functionality are straightforward, readily performed, and include extensive evaluation and penetration testing above and beyond what may be performed on voting systems that do not contain witness devices.

Responsible Entity: Testing Authorities
Process: Pre-Voting

Discussion: Witness device manufacturers will need to document their systems extensively and subject them to highly stringent testing.

4.3.4.8 Because voters must trust that the witness device records their verifications accurately, the results of witness system assessments are made available publicly.

Responsible Entity: Testing Authorities
Process: Pre-Voting

4.3.4.9 A voter should be able to inspect the record of the voter's verification upon the voter's request.

Responsible Entity: voting system vender
Process: Voting

Discussion: It is desirable that a voter has some capability to verify that the witness device is operating as specified.

4.3.4.10 The witness device clearly indicates any malfunction in a way that is obvious to poll workers and voters.

Responsible Entity: voting system vender, Voting Officials
Process: Voting

Discussion: This requirement serves to ensure that voting cannot continue if the witness device is not operating or malfunctioning.

4.3.4.11 The records captured by the witness device are able to be used in highly accurate audits of the voting records captured and stored by the recording station.

Responsible Entity: voting system vender
Process: Voting

4.3.4.12 The records contain unique identifiers that correspond to records stored by the recording station.

Responsible Entity: voting system vender
Process: Voting

4.3.4.13 The records are digitally signed by the witness device so that the integrity and authenticity of its records can be verified in audits.

Responsible Entity: voting system vender
Process: Voting

4.3.4.14 A witness device is able to export its records in an open, nonproprietary format such that the records can be used an automated audits.

Responsible Entity: voting system vender
Process: Voting

4.3.4.15 The records are stored in the witness device and exported such that voter privacy is protected, e.g., by making the order of the records randomly determined.

Responsible Entity: voting system vender
Process: Voting

4.3.5 End to End (Cryptographic) Independent Verification Systems (Informative)

This section contains very preliminary definitions for End to End (or cryptographic-based) independent verification systems. They are consistent with the definition of independent verification systems from Section 6.0 and build on the core definitions from Section 0.

End to end voting systems use cryptographic mechanisms as a substitute for some physical, computer-security, or procedural mechanisms used to secure other voting systems. Some auditing procedures normally performed by election officials at the tabulation center can done by voters or their designated representatives, using receipts issued by the voting system that work in conjunction with the cryptographic mechanisms. Several types of cryptographic voting schemes have been proposed or implemented, with varying properties. There are many cryptographic

techniques (such as secure multiparty computation and homomorphic) that could be applied in novel ways within future voting systems.

4.3.5.1 End to end systems use cryptographic mechanisms as a substitute for some physical, computer security, and procedural mechanisms used to secure voting systems. These mechanisms can be used by a voter to verify that ballot choices were recorded correctly and counted in the election.

Responsible Entity: voting system vender
Process: Voting

Discussion: There are potentially many types of end to end systems that could perform a variety of different functions.

4.3.5.2 End to end systems record voters ballot choices at an electronic voting system and encrypt the records of votes for later counting by designated trustees.

Responsible Entity: voting system vender
Process: Voting

Discussion: The voting station would operate much as a DRE.

4.3.5.3 End to end systems produce a receipt that can be used by the voter in some process made available by election officials so that the voter may verify that the voter's ballot choices were recorded correctly and counted in the election.

Responsible Entity: voting system vender
Process: Voting

Discussion: The receipt could have a variety of different forms but likely would be printed on paper for the voter's ease of handling.

4.3.5.4 No one trustee is able to decrypt the records; decryption of the records is performed by a process that involves multiple trustees.

Responsible Entity: voting system vender, Voting System Officials
Process: Post-Voting

Discussion: For example, multiple keys could be combined to decrypt the records.

4.3.5.5 The receipt preserves voter privacy by not containing any information that can be used to show the voter's choices.

Responsible Entity: voting system vender
Process: Voting

4.3.5.6 The process used to verify that ballot choices were recorded correctly or counted in the election preserves voter privacy by not revealing any information that can be used to show the voter's choices.

Responsible Entity: voting system vender
Process: Voting

4.3.5.7 End to end systems store backup records of voter's ballot choices that can be used in contingencies such as damage to or loss of its counted records.

Responsible Entity: voting system vender
Process: Voting

Discussion: This is necessary because the handling of the encrypted records requires the same chain of custody procedures as records produced by other voting systems and are thus subject to loss or damage. This could be paper for example.

4.3.5.8 The backup records contain unique identifiers that correspond to unique identifiers in its counted records, and the backup records are digitally signed so that they can be verified for their authenticity and integrity in audits.

Responsible Entity: voting system vender
Process: Voting

4.3.5.9 Cryptographic software in end-to-end systems is documented thoroughly, subject to extensive verification testing for correctness. The documentation includes extensive discussion of how cryptographic keys are to be generated, distributed, managed, used, certified, and destroyed.

Responsible Entity: Testing Authorities
Process: Pre-Voting

Discussion: The correctness of the system depends on the correctness of the cryptographic algorithms and their implementations. Thus, rigorous testing is necessary.

4.3.5.10 Vote capture stations used in end to end systems meet all security, usability, and accessibility requirements for similar stations in other voting systems.

Responsible Entity: voting system vender
Process: Voting

4.3.5.11 Reliability, usability, and accessibility requirements for printers in other voting systems apply as well to receipt printers used in end to end systems.

Responsible Entity: voting system vender
Process: Voting

4.3.5.12 Trustee systems are subject to the same evaluations and assessments as other voting systems.

Responsible Entity: voting system vender
Process: Pre-Voting

4.3.5.13 Systems for verifying that voters' ballots were recorded properly and counted in the election are implemented in a robust secure manner.

Responsible Entity: Voting System
Process: Post-voting

Discussion: Many of the cryptographic schemes have a "public append-only bulletin board" as a component; this is an important part of the system and needs to be implemented in a robust secure manner.

5 System Testing Program

In accordance with HAVA Section 231(b), laboratories that test and certify voting systems must be accredited by the EAC. In most cases, this will be a two-step process.

Step 1: The Director of NIST conducts an evaluation of independent, non-Federal laboratories to determine their competence to conduct tests of voting systems under HAVA provisions. This evaluation will be accomplished through the accreditation of qualified laboratories for testing of voting systems by the National Voluntary Laboratory Accreditation Program (NVLAP). Qualified laboratories are called Voting System Testing Laboratories (VSTL). The Director of NIST submits a list of independent, non-Federal VSTLs to the EAC, per Section 231(b)(1).

Step 2: The EAC considers the recommendations of the NIST Director and any additional criteria established by the EAC and votes to accredit testing authorities according to Commission rules. This authorizes EAC-accredited testing authorities to test, certify, decertify, and recertify voting systems under HAVA.

The NIST Director has chosen to utilize NVLAP accreditation as the means to conduct the evaluation to qualify laboratories for the list submitted to the EAC as required by Section 231(b)(1). Per Section 231(b)(2)(B), Step 1 is not explicitly a pre-condition for Step 2, because the EAC has the authority to accredit labs not on the NIST Director's list.

The EAC can define the conditions and requirements for EAC accreditation of testing authorities. The basic requirement would be compliance with EAC policies and procedures for certification, decertification, and recertification. The complexity of those policies and procedures, and the means for determining compliance with them, will depend on EAC's intentions for the process. The simplest approach would be an attestation from the laboratory that it will comply with EAC requirements. This would provide a means for EAC to suspend or revoke an EAC accredited testing authority's accreditation for issues related to interaction with the EAC. When such issues are not related to the lab's competence to perform tests, its NVLAP accreditation status may not be affected.

In accordance with HAVA, the EAC-accredited testing authority will initially test systems to the 2002 Voting System Standards until such time as the EAC adopts new voting system standards.