ClearVote 2.1

ClearVote Configuration Management Plan

ClearVote Configuration Management Plan

Clear Ballot Part Number: 100057-10017

Clear Ballot Group
2 Oliver Street, Suite 200
Boston, MA 02109
857-250-4961
clearballot.com

Document history

Date	Description	Version	Authors
01/10/2017	Initial submission to EAC	1.0	Joni G. McNutt
02/03/2017	Minor typographical and reference-related edits	1.0.1	Joni G. McNutt
03/30/2017	Minor typographical and reference-related edits based on feedback from the state of Colorado	1.0.2	Joni G. McNutt
04/28/2017	Minor updates based on feedback from the State of Colorado and Clear Ballot Quality Assurance	1.0.3	Joni G. McNutt
06/16/2017	Minor updates for vote-by-mail campaign	1.0.4	Joni G. McNutt
10/04/2017	Updated the Hardware naming conventions, Maintaining components until retirement, Establishing a baseline for a document, Initial release scenario to an accredited test lab, Upgrade releases, Supported hardware, Required software, and Configuration management resources sections; minor edits	1.0.5	Joni G. McNutt
10/17/2017	Updated the Defect reporting, the Functional configuration audit, and the Maintaining components until retirement sections; added reference to Baseline and promotion activities chapter	1.0.6	Joni G. McNutt
10/27/2017	Updated the Maintaining components until retirement, Hardware, and Upgrade release sections; minor edits	1.0.7	Joni G. McNutt
11/11/2017	Added VVSG test version determination section to Baseline and promotion activities chapter	1.0.8	Joni G. McNutt
01/19/2018	Vote-by-Mail campaign 2	1.0.9	Joni G. McNutt
04/13/2018	Minor edits	1.0.10	Joni G. McNutt
06/01/2018	Updated cover, added Fujitsu fi-6670 information	1.0.11	Joni G. McNutt
06/15/2018	Updated cover, removed Fujitsu fi-6670 information	1.0.12	Joni G. McNutt
08/03/2018	Added encrypted USB drives to Required Accessories section, minor edits	1.0.13	Joni G. McNutt
08/15/2018	Updated cover	1.0.14	Joni G. McNutt

Date	Description	Version	Authors
11/06/2018	Minor edits	1.0.15	Joni G. McNutt
04/12/2019	Minor edits	1.0.16	Joni G. McNutt
11/04/2019	Minor edits	1.0.17	Joni G. McNutt
02/12/2020	Minor edits	1.0.18	Joni G. McNutt

Table of contents

Preface

This section defines the purpose of this document. It contains the following subsections.

- About this document
- Scope of this document
- Intended audience

About this document

This document provides information about Clear Ballot's configuration management program for the ClearVote suite of products.

It corresponds to the *VVSG Volume 2 Section 2.11* requirement for the technical data package (TDP).

> ✓ A ClearVote® system can comprise the ClearAccess®, ClearAudit®, ClearCast®, ClearCount®, and ClearDesign® products. Jurisdictions are not required to purchase all products. You can ignore references to any ClearVote products that are not part of your voting system. Also ignore implementation options that are not relevant to your policies and procedures.

Scope of this document

This document provides information about the following aspects of ClearVote configuration management:

- Configuration management policy
- Configuration identification
- Baseline and promotion activities
- Configuration control procedures
- Release process
- Configuration audits
- Configuration management resources

Intended audience

This document is intended for state and federal election officials and their voting system test laboratories as part of the technical data package (TDP) required to certify the ClearVote voting system for use. This document is also used by Clear Ballot personnel who support election officials and election staff.

Chapter 1. Configuration management policy

Responsive to VVSG 2005, Volume 1, Section 9.2.

Clear Ballot incorporates configuration management best practices throughout its development cycle. The ClearVote configuration management policy applies to all ClearVote systems and components to ensure end-to-end process and product control, and consists of the following features:

- Software tools for version control and defect tracking
- Configuration management processes as outlined in this document
- A quality assurance (QA) team that reports to the vice president of engineering

Chapter 2. Configuration identification

Responsive to VVSG 2005, Volume 1, Section 9.3.

This section describes ClearVote classification, versioning, and naming conventions.

2.1   Identifying discrete system components

Responsive to VVSG 2005, Volume 1, Section 9.3.1.

The *ClearVote System Overview* introduces the suite of ClearVote products. For more details about the ClearAccess, ClearCast, ClearCount, and ClearDesign systems, see the system overview for each product.

The *ClearVote Personnel Deployment and Training Plan* describes training requirements for each product in detail.

For the purposes of configuration management, ClearVote software comprises:

- ClearAccess
- ClearCast
- ClearCount
- ClearDesign

For example, a ClearCount release consists of the correctly identified versions of the following elements:

ScanServer computer

The ClearCount software installation package (which includes the required Linux operating system and third-party software)

ScanStation computers

- Microsoft Windows 10 Pro operating system
- Fujitsu TWAIN driver
- Fujitsu ScandAll PRO software

Election administration station computers

- Microsoft Windows 10 Pro operating system
- Browser (Google Chrome)

2.1.1 Filenaming conventions

The products comprising the ClearVote system use the filenaming conventions described in this section.

2.1.2 Hardware naming conventions

The ClearVote system uses all commercial off-the-shelf (COTS) hardware. Therefore, Clear Ballot does not name said hardware; it refers to existing hardware names and serial numbers as described in this section.

The components comprising the ClearCast system are also COTS hardware, with the exception of the sheet-metal enclosure that allows those components to be assembled together as a voting station. Clear Ballot lists these components in a manufacturing bill of materials for each ClearCast hardware revision, and like the discrete COTS parts used in other ClearVote system components, does not rename them.

In an election environment, each ScanStation computer is identified with the scanner model, the serial number of the computer, and the name of the computer. This information, along with the election name, appears in the Tabulator window on that ScanStation computer. The following is an example of an election and ScanStation name:

ClearCountyGeneral12 – FUJITSU fi-6800 – 00081 – SCANNER_06

2.1.3 Software filenaming conventions

The ClearVote software that is written in Python follows the filenaming conventions established in the PEP-8 standard. Software that uses JavaScript conforms to Crockford Style industry-standard filenaming conventions, which is documented on web sites such as http://javascript.crockford.com/code.html.

2.1.4 Documentation filenaming conventions

Documentation files are named as follows:

<product name> <document name> <mmddyy>.pdf

For example:

- ClearCount Election Administration Guide 072519.pdf
- ClearCast Poll Worker Guide 072519.pdf

The *Document history* section at the beginning of each document indicates the date, the version, the author and the changes made.

2.2 Versioning conventions

Responsive to VVSG 2005, Volume 1, Section 9.3.2.

The major software components of the ClearVote product suite—ClearAccess, ClearCast, ClearCount and ClearDesign—may have different software version and release numbers. However, within a major software component, all modules use common software version and release numbers.

2.2.1 Major version releases

A major release is one that consists of major changes that alter or add to the functionality of the product and change the way in which the user interacts with it. These changes are rigorously tested to ensure proper functionality of new features, as well as to prevent any regression of existing features. Each major release also includes additional documentation that supports new features and instructs users about system changes.

For a major release, the version number of the product package is incremented before the decimal place. For example, an increase from version 1.5 to version 2.0 indicates a major release.

2.2.2 Minor version releases

A minor release consists of small changes that make minor improvements or a collection of bug fixes to the code that improve or correct the underlying functionality of the product without changing the user experience. These changes are rigorously tested to ensure they have the proper effect on the product without changing the user experience.

For a minor release, the version number of the product package is incremented after the decimal place. For example, an increase from version 1.4 to version 1.5 indicates a minor release.

2.2.3 Bugfix releases

A bugfix release consists of a time-critical bug fix required to allow a customer to continue using the product in the event that a critical bug is found at a customer site.

For a bugfix release, the version number of the product is incremented with a second decimal place. For example, an increase from version 1.4 to version 1.4.1 indicates a bugfix release.

Chapter 3. Baseline and promotion activities

Responsive to VVSG 2005, Volume 1, Section 9.4, and EAC Decision on RFI 2012-03 (Configuration Management of COTS Products).

The following section describes baseline and promotion activities for the ClearVote software, hardware and their configurations, as well as for the ClearVote product documentation. It also contains the Clear Ballot COTS management plan.

3.1 Establishing a baseline for a component

Clear Ballot defines a formal or starting baseline for a new component at the point when that component is ready to enter the internal test process.

3.2 Promoting a subsequent version to baseline status

Clear Ballot uses an internal versioning scheme based upon its source control system.

When Clear Ballot releases a version to its voting system test laboratory (VSTL), it gives that version a three-part version number. If updates are needed in response to test findings, the third part of the version number is incremented to reflect the change.

Clear Ballot updates the full version number for each new certification.

After establishing a starting baseline for a component, Clear Ballot promotes subsequent instances of the component to baseline status as development progresses. When components are modified or updated, changes to associated software or documentation are checked into the version control system with a specific tag.

Clear Ballot tags versions in source control. Clear Ballot records the tag each time a version is created and given to an outside party. The tag and action are recorded in source control.

3.3 Maintaining components until retirement

Clear Ballot keeps track of every source file intended for release during its entire lifecycle. The files that comprise every component in the shipped product are kept under the source control system. Clear Ballot observes version control best practices for the life of the configuration item. (The life of a configuration item begins with its creation and ends when it becomes unnecessary or obsolete.)

Version control best practices followed by the Clear Ballot development organization include the following:

- Committing related changes together, and, conversely, not committing unrelated changes in the same action
- Committing completed units of work

- Committing frequently

- Testing code prior to committing

- Coordinating and sharing changes with other software engineers working on related code

- Providing explicit and descriptive commit messages along with the code

- Ensuring all software engineers and test analysts follow these guidelines, by sharing best practices and monitoring conformance

For hardware components, the Clear Ballot Product Management team maintains visibility of the various COTS products' lifecycles. Clear Ballot seeks COTS products with longer lifecycles, which tends to follow the need for commercial/industrial/military electrical and mechanical robustness.

COTS products require management of their lifecycles, particularly the manufacturers' end of life, general lifecycle length, and policies for successor products. Clear Ballot's hardware engineer researches specifications and sourcing, and evaluates COTS candidates to ensure that *VVSG* and Clear Ballot requirements are met. The hardware engineer manages and tracks all COTS components via the *ClearVote Product Ecosystem* database.

Product Management uses the *ClearVote Approved Parts List* to track specific key components' end-of-life. "COTS management" in the *ClearVote Approved Parts List* includes the expected lifecycle length for all categories of COTS items.

As currently approved COTS components become end-of-life at the manufacturer, Product Management evaluates newer COTS devices via specification sheets, and then through testing. When a candidate device passes QA testing, Clear Ballot places it on the *ClearVote Approved Parts List* for a given revision of the ClearVote system.

3.4   Establishing a baseline for a document

Clear Ballot defines a formal, or starting, baseline for a document when the document is ready to enter the review process. At the start of the process, documents are versioned 1.0.

Every document required for the ClearVote TDP contains a *Document history* section. The history includes the author, date, and scope of changes, as well as a corresponding version number.

3.5   *VVSG* test version determination

Clear Ballot's Engineering and Product Management teams evaluate systems as their release content and timing is planned to determine to which *VVSG* standards each system will be tested. Systems will be tested to a later *VVSG* standard than their baseline certification when one of the following conditions occurs:

- Fifty percent of functional lines of source code are changed

- There exists a new voter-facing vote-capture device in the submitted system to the EAC

- There exist architectural changes that introduce significant new security vulnerabilities

- There is consensus among Clear Ballot Engineering and Product Management teams that the system has strayed so far afield of the configuration certified to an earlier *VVSG* standard that either:

    - It would be a poor practice to continue baseline testing to that older *VVSG* standard, or

    - The VSTL will be directed to perform baseline testing on the system to be submitted even though none of the conditions described above has occurred.

Chapter 4. Configuration control procedures

Responsive to VVSG 2005, Volume 1, Section 9.5.

Clear Ballot controls and logs access to configuration items to prevent unauthorized or untracked additions, changes, or deletions.

Clear Ballot access control is monitored by the DevOps team, who grants credentials to Clear Ballot employees based upon their job functions. All ClearVote software and documentation is kept under source control, and each change is identified by the person who made the change. Client connections to the Clear Ballot secure server that hosts the ClearVote software and documentation are made via SSH keys, which require users to log in using credentials granted by the DevOps team.

4.1   Developing and maintaining internally developed items

Clear Ballot maintains configuration control over its internally developed items, as well as any third-party dependencies, such as the Python runtime environment, through source control and change control.

4.1.1 Source control

Changes to software modules and associated data and documentation files are tracked using a distributed source control management system (SCMS) called GitLab, which accomplishes the following goals:

- Maintains the deltas (changed contents) of each revision of each source file (code or documentation)

- Maintains history of all code or documentation changes of individual files within the system. This history can be tracked by following the SCMS comment log.

- Maintains a list of all files that make up the source files for each revision to the system

- Manages all documentation iterations. As documentation is developed and submitted for technical review, it is stored in the SCMS, and tracked continuously through its revisions.

Full documentation for the GitLab SCMS and other tools is listed in "Configuration management resources" on page 28.

Changes to source code and documentation are always logged into the SCMS, along with the author and time, to create a proper record of the evolution of the product. The SCMS repositories maintain permissions to ensure all changes are made with appropriate internal approvals before they can be merged into the main source stream. The lead software engineer reviews and approves all changes before they are merged.

Clear Ballot uses the access control system of GitLab to control modification access to source files. GitLab provides remote read/write access to a centralized repository using SSH public key authentication.

4.1.2 Change control

Using its SCMS, the Clear Ballot software engineering team employs a master repository of sources. Each software engineer also uses software engineer-specific repositories for his or her work in progress. When a feature is completed, the changes in the software engineer-specific repositories are merged into the master repository, each change awaiting approvals to be merged into the main branch of the product.

Development for a next major release occurs within a single main branch in the master repository. When development completes on a specific released version of the software, and all changes have been reviewed by the lead software engineer for release, the source files for the main branch are tagged to permanently record the exact list and contents of the files that compose that released version. A side branch is then created for any critical fixes to that released version. Critical fixes are only permitted into the side branch of a released version if they are required to maintain Clear Ballot's customers' ability to use that version, and only as approved by the vice president of engineering.

4.2 Acquiring and maintaining third-party items

The third-party items included in the ClearVote submission for certification comprise these categories:

- COTS hardware that runs the ClearVote software
- Open-source software artifacts included in the ClearVote system

4.2.1 Hardware

The products that comprise the ClearVote system run on COTS computers, scanners, printers, and network switches. The only exception is the ClearCast sheet-metal enclosure that allows the ClearCast COTS components to be assembled together as a voting station. The design and specification of the sheet-metal enclosure are under the control of Clear Ballot.

Jurisdictions purchase this COTS hardware directly with reference to requirements provided by Clear Ballot. For detailed information about hardware, see the ClearAccess, ClearCast, ClearCount and ClearDesign hardware specifications.

4.2.2 Software

All of the third-party software artifacts included with the ClearVote system can be freely obtained and updated using the Internet, and are maintained by their respective authors. All third-party software included in the ClearCount product of the ClearVote system is unmodified.

Clear Ballot downloads the versions it requires and then stores and tracks the software in its source control system, as described in "Source control" on the previous page. The SCMS maintains a record of the software that was used in each build.

Clear Ballot also obtains the required versions of scanner software and drivers, and stores said versions in its source control system. Clear Ballot then distributes these to the test laboratory or to customers.

Due to size and licensing constraints, Clear Ballot does *not* store the required operating system software in its SCMS repository.

For details of the software tools used to develop the ClearVote products, see the software design specification for each product.

4.3 Resolving defects

Clear Ballot uses an issue-tracking system called Jira to log and track tasks, enhancement requests, and defects. The Clear Ballot implementation of Jira uses the following fields:

Field	Description
Number	The ticket number, which serves as a unique identifier for an issue
Project	A designation of the area impacted by the issue (such as, software development, documentation, hardware development)
Type	Options of task, subtask, epic, story, bug, or enhancement
Summary	A brief description of the issue that serves as a title
Description	A detailed description of the issue
Created	The date the issue was created
Team	The team assigned to the issue
Due date	The date by which issue resolution needs to occur
Priority	The priority assigned to this issue (highest, high, medium, low, lowest)
Components	The ClearVote system components involved in the issue
Affects version	The product version impacted by the issue
Fix version	The target version for the issue resolution
Sprint	A nonrelease target for the issue resolution
Reporter	The Clear Ballot team member who opened the issue
Assignee	The Clear Ballot team member assigned to the issue
Status	The status of the issue (new, in progress, review, blocked, backlog, closed)
Comments	Additional notes and updates concerning the issue
Resolution	A designation of how the issue was resolved (done, invalid, won't do)

Depending upon the regulations in the jurisdiction, any bugfix release may need to be certified before being released to customers.

4.3.1 Resolving internally identified defects

The Clear Ballot workflow for resolving internally identified defects is as follows:

1. While testing or otherwise working with the software, a Clear Ballot team member, such as a QA analyst, uncovers a potential defect.

2. The Clear Ballot team member creates a ticket in the Jira issue-tracking system.

3. The Clear Ballot scrum team assigns a priority ranking to the issue and assigns it to a team member for resolution.

4. The Clear Ballot team member assigned to the issue works toward the issue's resolution.

5. After the issue is fixed, Clear Ballot incorporates the fix into upcoming releases. A Clear Ballot team member (frequently a QA analyst) verifies the fix in-house prior to release. Depending upon the severity and pervasiveness of the problem, Clear Ballot might also inform customers of the issue and provide a bugfix release (bearing in mind that any code changes would need to be recertified).

4.3.2 Resolving externally identified defects

The Clear Ballot workflow for resolving externally identified defects is as follows:

1. A customer encounters and reports an issue to Clear Ballot by one of the following methods:

   - Through the Clear Ballot customer portal
   - By contacting a Clear Ballot customer support specialist via telephone or e-mail

2. A Clear Ballot team member creates a ticket in the Jira issue-tracking system, as well as in the customer relationship manager application.

3. The Clear Ballot scrum team assigns a priority and severity ranking to the issue and assigns it to a Clear Ballot team member for evaluation and resolution.

4. The Clear Ballot team member assigned to the issue evaluates it. The subsequent approach depends upon the nature and severity of the issue. Possible outcomes include:

   - Issue to be resolved in the next planned release
   - Issue to be resolved in a bugfix release
   - Issue to be resolved in a workaround
   - Not a bug/Product works as intended

5. The Clear Ballot team works toward the resolution of any verified issue. Concurrently, a Clear Ballot customer support specialist keeps the customer apprised of changes in the issue's status.

The severity level of an issue is based upon its impact upon the customer, as outlined below:

Table 4-1. Issue severity levels

Severity level	Description of impact
Highest	The issue causes the voting system to crash, stop functioning, lose or misinterpret data, or otherwise bring into question election integrity, accuracy, or security.
High	The issue, while not fatal, affects a major part of the system.
Medium	The issue, while significant, affects a noncritical part of the system.
Low	The issue affects a minor part of the system, or has little impact on its operation.
Lowest	The issue is unlikely to occur or have any impact upon operations

Chapter 5. Release process

Responsive to VVSG 2005, Volume 1, Section 9.6.

This section describes the ClearVote release process for the test laboratory and for customers.

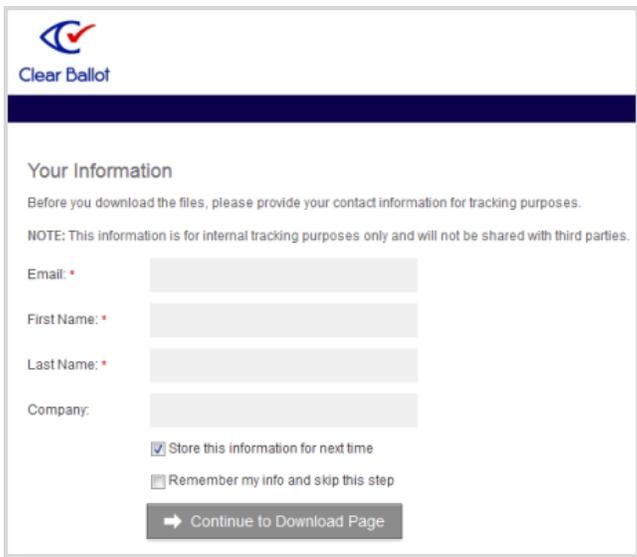## 5.1   Initial release scenario to an accredited test lab

For test campaigns relating to federal certification, Clear Ballot delivers all hardware, software, build methods, documentation, and any accessories or tools needed directly to the VSTL. For state campaigns where the state relies on VSTL testing, Clear Ballot submits all of the artifacts (software, hardware, and documentation) necessary for testing to state election officials. They, in turn, forward these materials to the test laboratory.

5.2   Release process example: ClearVote

When a ClearVote package is ready to download, Clear Ballot sends an e-mail to the applicable election officials. This e-mail message contains a download link to the secure download utility, as well as an SHA-256 digest for each file to be downloaded. These digests, which consist of a string of letters and numbers and the filename, can be used with any SHA-256-compliant hash tool to validate the files. Any discrepancies should be reported to Clear Ballot immediately.

The procedure to download a ClearVote package is as follows:

1. Click the link in the e-mail from Clear Ballot.

2. Sign in on the **Your Information** page and click **Continue to Download Page**.

3. On the Download page, locate the ClearVote file and click **Download**.

4. In the Opening dialog box, select **Save File** and click **OK**.

5. Close the browser.

Upon receipt, the test laboratory installs the software as described in the applicable product's installation guide.

5.3 Upgrade release scenario

The test laboratory can reinstall a ClearVote software package at any time without loss of configuration settings. Running the previously used installer package can repair damaged files. Running a new installer package updates the system. It is not necessary to uninstall the prior version before installing the new version.

5.4 Initial release scenario to jurisdictions

Jurisdictions obtain the ClearVote DVDs from their state election officials. For details about how a customer installs the software, see the installation documentation for each ClearVote product.

5.5 Upgrade releases

Software: Release of software upgrades to customers depends upon individual state requirements. If a state allows a jurisdiction to receive software from the system manufacturer, Clear Ballot copies the software from its internal repository to DVD and sends it via express courier to the jursidiction. Processes, such as the application of tamper-evident seals and the use of courier tracking numbers, are employed to help ensure the security of the software in transit. If a state requires that the software come from the VSTL or the EAC, Clear Ballot or that governmental body requests the delivery of the software.

Hardware: COTS components for upgrade can be ordered by the customer from the manufacturer in consultation with Clear Ballot, or directly from Clear Ballot, based on contractual arrangements for that customer. Regardless, the *ClearVote Approved Parts List* governs what is ordered and subsequently sent to the customer. Customers receive a copy of the *ClearVote Approved Parts List*, which includes the expected lifecycle length for all categories of COTS items.

Documentation: Clear Ballot staff, typically in Operations or Customer Support, supplies documentation associated with any field upgrades.

For details of how to upgrade software for the ClearVote products, see the installation guides for each product.

5.6   Verifying the version of the software in a fielded version

The ClearCount system includes reports that verify the installed product software version. Log in to an election administration station and then select the **About This Software** option from the user name drop-down list at the upper right. The version number appears at the upper left of the ClearBallot Product Files and Installed System Packages reports.

> Together, these reports list all software packages installed on the ScanServer computer. For details, see the *ClearCount System Identification Guide* and "Obtaining the list of all software files present on the system" in the *ClearCount Election Preparation and Installation Guide*.

In the ClearDesign and the ClearAccess systems, the installed product software version number appears at the bottom of the main window. On the ClearCast voting station, log in with the maintenance role and select **Settings> About ClearCast**. See the *ClearAccess System Identification Guide*, the *ClearCast System Identification Guide*, and the *ClearDesign System Identification Guide* for additional information.

Chapter 6. Configuration audits

Responsive to VVSG 2005, Volume 1, Section 9.7.

There are two required configuration audits:

- Physical configuration audit (PCA)
- Functional configuration audit (FCA)

6.1 Physical configuration audit

Responsive to VVSG 2005, Volume 1, Section 9.7.1.

In the PCA, the VSTL compares the ClearVote system components to the ClearVote technical documentation. The test laboratory assembles its own version of the ClearVote configuration baseline, including hardware setup and software installation, using the documentation provided.

The specification for the ClearVote system is described in the following sections.

6.1.1 Supported hardware

The ClearVote system uses COTS hardware exclusively. For details, see the *ClearVote Approved Parts List*.

The components comprising the ClearCast system are also COTS hardware, with the exception of the sheet-metal enclosure that allows those components to be assembled together as a voting station.

6.1.2 Required software

For details of third-party software, see the *Build Procedures* for each product in the ClearVote system.

Clear Ballot does use open-source software, but no other proprietary third-party software. All API libraries are open source. The programming language used to create the ClearVote software, Python, is not a compiled language. Therefore, no compiler is needed. To construct run-time executables, Clear Ballot uses the PyInstaller tool.

6.1.3 Documentation

The ClearVote product documentation is provided on a disc that accompanies the installation DVD.

6.1.4 User acceptance test procedures and acceptance criteria

For details of acceptance test procedures and criteria, see:

- *ClearAccess Acceptance Test Checklist*
- *ClearCast Acceptance Test Checklist*
- *ClearCount Acceptance Test Checklist*
- *ClearDesign Acceptance Test Checklist*

6.1.5 Comparison between the PCA and FCA

Currently, there are no changes between the system used for the PCA and the system used for the FCA.

6.2 Configuration baseline

This section provides details about how the hardware must be set up for the configuration audit. It is based on the recommended minimal setup of a single ScanStation computer with an attached scanner.

6.2.1 Hardware for a configuration audit

Hardware required for a configuration audit includes the following:

Product	Item
ClearAccess	One touchscreen computer
	One printer
ClearCast	One ClearCast voting station with ballot receptacle
ClearCount	One scanner: Fujitsu fi-6400, Fujitsu fi-6800, fi-7800, fi-7900, or Fujitsu fi-7180
	Scanners: ibml ImageTracDS 1155 (or ibml ImageTracDS 1210 or ibml ImageTrac 6000 series)
	One ScanServer computer
	One ScanStation computer
	One election administration station computer
	One network switch
ClearDesign	One DesignStation computer
	One DesignServer computer
	One network switch

See the *ClearVote Approved Parts List* for a list of approved components.

6.2.2 Firmware/Software configuration

Software required for a configuration audit includes the following:

- ClearAccess: Operating system and ClearAccess application software
- ClearCast: Trusted build for ClearCast, which includes the operating system, ClearCast application software, and any needed third-party packages
- ClearCount:
  - Microsoft Windows operating system for ScanStation and election administration station computers
  - Fujitsu drivers and firmware
  - ScandAll PRO software
  - Google Chrome browser
  - ClearCount software (includes the Ubuntu Linux operating system)
- ClearDesign: Operating system, ClearDesign application software, and Google Chrome browser

For further details, see the installation documentation for each product in the ClearVote system.

6.2.3 Required accessories

Accessories required for a configuration audit include the following:

Table 6-1. Required accessories

Item	Version
Network switch	Gigabit network switch (for example, Cisco SG250)
Cables	Three Cat-5 Ethernet cables, plus one additional Ethernet cable for each additional ScanStation computer
USB drives	Two USB drives (for example, SanDisk Ultra Flair 32 GB)
Keypad	Accessible keypad (for example, Storm Interface EZ Access keypad)
Printer	Laser printer (for example, Brother HL-2350DW)

See the *ClearVote Approved Parts List* for a list of approved accessories.

6.2.4 Required settings

For the lists of required settings, see the installation documentation for each product in the ClearVote system.

6.3 Functional configuration audit

Responsive to VVSG 2005, Volume 1, Section 9.7.2.

In the functional configuration audit, the VSTL performs all the functions described in the system documentation. Clear Ballot's VSTL (Pro V&V) has comprehensive test plans that describe the procedures it uses to support this audit.

Prior to release to the VSTL, Clear Ballot performs a functional configuration audit that includes the following:

- Installation
- Ballot design and printing
- BDF and ADF files
- Scanning/Tabulation on supported scanners
- Accessible voting
- Precinct voting and merging results
- Operational reporting
- Adjudication and resolution
- Results reporting (PDF) and exporting (XML and CVR)
- Log file conformance
- End-to-end testing for each election type
- Security, including SCAP
- Performance (throughput)
- System limits
- Regression tests
- Documentation conformance
- *VVSG* compliance

The 9000-plus documented test cases that comprise Clear Ballot's functional configuration audit are archived as described in the *ClearVote Test and Verification Specification*.

6.4   Defect reporting

During the development phase, all defects encountered are reported and tracked through resolution in Clear Ballot's Jira issue-tracking system. This system is described in the *ClearVote Test and Verification Specification*.

After product release, any problems encountered in the field are reported and tracked in Clear Ballot's customer help desk system. The Clear Ballot Customer Success team investigates each report by involving resources in the Engineering, QA, Documentation, and Product Management teams as needed. If the problem cannot be resolved in the field, Clear Ballot issues a field service bulletin to alert customers to the existing problem while updates to the product are in progress.

If the affected product carries a federal certification, the Clear Ballot's Certification department also notifies the EAC, in keeping with their processes.

Chapter 7. Configuration management resources

Responsive to VVSG 2005, Volume 1, Section 9.8.

Clear Ballot handles configuration management using the open-source tool called GitLab. Clear Ballot uses GitLab to manage:

- Software source files

- TDP documents

GitLab is a distributed version control system. The nature of distributed version control systems such as GitLab is that processing occurs on individual client machines, *not* on a central server. (In fact, GitLab does not require a central server.)

GitLab is supported on all operating systems. The operating environment for the tool is the same as the operating system on a client. Clear Ballot software engineers use any of the Windows, Linux, or MacOS systems, and install the latest version of GitLab for the local operating system. GitLab is always installed in its default program folder.

For detailed information about GitLab procedures and training materials, see
https://about.gitlab.com/product/.

For information about Clear Ballot's GitLab processes, see "Configuration control procedures" on