ClearVote 2.1

ClearCount Software Design and Specification

ClearCount Software Design and Specification

Clear Ballot Part Number: 100019-10017

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 updates 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
06/30/2017	Restructured sections 3.4.1, 3.4.2 and 3.4.3 to include sections for software items, configuration data files and procured files (added sections 3.4.2.3 and 3.4.3.3); rearranged table 3-2; updated the table in section 3.4.2.1; updated table 3.10 and added versions	1.0.5	Joni G. McNutt
11/01/2017	Updated the Interface identification section, minor edits	1.0.6	Joni G. McNutt
11/11/2017	Updated the Software environment section, added graphic to Tabulator component section, added graphic to Server component section	1.0.7	Joni G. McNutt
01/19/2018	Vote-by-Mail campaign 2	1.0.8	Joni G. McNutt
04/27/2018	Minor edits	1.0.9	Joni G. McNutt
06/01/2018	Updated cover, added Fujitsu fi-6670 information	1.0.10	Joni G. McNutt
06/15/2018	Updated cover, removed Fujitsu fi-6670 information	1.0.11	Joni G. McNutt
08/15/2018	Updated cover, minor edits	1.0.12	Joni G. McNutt
04/12/2019	Added information about encrypted BDFs, minor edits	1.0.13	Joni G. McNutt
11/22/2019	Minor edits	1.0.14	Joni G. McNutt
02/12/2020	Minor edits	1.0.15	Joni G. McNutt

Table of contents

Preface **7**

Chapter 1. High-level system component overview **8**

1.1 Applicable documents 9

Chapter 2. Security overview **10**

2.1 User access levels 10

2.2 MySQL user identification 10

2.3 Election database phases 11

2.3.1 Election database lifecycle 11

2.4 Artifacts 12

2.4.1 ScanStation artifacts 13

2.4.2 ScanServer artifacts 13

Chapter 3. Software overview **16**

3.1 ScanServer 16

3.1.1 CIFS 16

3.1.2 MySQL 16

3.1.3 Apache web server 17

3.1.4 ClearCount web server 17

3.1.5 ScanServer directory structure 17

3.2 ScanStation 18

3.2.1 COTS scanner software 19

3.2.2 The Tabulator application 19

3.2.3 The DeleteBox utility 19

3.3 Election administration stations 19

3.3.1 The Dashboard 21

3.4 Software item identification 22

3.4.1 ScanStation software 22
3.4.2 ScanServer software 27
3.4.3 Election administration station software 29
Chapter 4. Software standards and conventions 34
4.1 Software development methodology 34
4.2 Software design standards 34
4.3 Software specification standards 34
4.4 Software coding standards 34
4.5 Testing and verification standards 35
4.6 Quality assurance standards 35
Chapter 5. Software operating environment 36
5.1 Hardware environment and constraints 36
5.2 Software environment 36
Chapter 6. Software functional specification 37
6.1 Configurations and operating modes 37
6.1.1 Preparing the systems 37
6.1.2 Setting up for an election 37
6.1.3 Recording and tabulating cards 41
6.1.4 Resolving card tabulation problems 41
6.1.5 Closing the polling place 43
6.1.6 Closing the central count for an election 43
6.1.7 Generating reports 43
6.2 Software functions 43
6.2.1 Exception conditions 43
6.2.2 System failures 43
6.2.3 Data input/output errors 44
6.2.4 Error logging for audit record generation 44
6.2.5 Production of statistical ballot data 44

6.2.6 Security monitoring and control 44

Chapter 7. Programming specifications 46

7.1 Programming specifications overview 46

7.2 Tabulator application (tabulate folder) 46

7.3 Server component (server folder) 48

7.4 Package component (package folder) 49

Chapter 8. System database 50

Chapter 9. Interfaces 51

9.1 Interface identification 51

9.2 Interface description 52

9.2.1 ScanServer interface description 52

9.2.2 Apache interface 53

9.2.3 ClearCount web server 53

9.2.4 CIFS server components 54

9.2.5 ScanStation interface description 55

9.2.6 Election administration station interface description 64

9.2.7 ScanServer setup interface 65

9.2.8 Network switch interface 66

9.2.9 Election administration station-to-ScanServer interface for creating an election 66

9.2.10 ScanStation-to-ScanServer interface for ballot scanning 67

9.2.11 ScanStation-to-ScanServer interface for deleting tabulation results 67

9.2.12 Election administration station-to-ScanServer Card Resolutions interface 68

9.2.13 Clear Ballot HTTP requests 68

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 the physical and functional components of the ClearCount system, including how the components are organized and the software interfaces between them. Sections in this document correspond to requirements for the technical data package specified in *VVSG 2005, Volume 2, Section 2.5* as noted.

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 the ClearCount system:

- High-level system component overview
- Security overview
- Software overview
- Software standards and conventions
- Software operating environment
- Software functional specification
- Programming specifications
- System database
- Interfaces

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. High-level system component overview

A ClearCount system is a network of computers communicating over a closed, hardwired Ethernet. Connected computers include the following:

ScanServer

There is one server running Ubuntu Linux. Running on the server are:

- MySQL database server—All of the following system databases are MySQL databases:
  - mysql—Used by MySQL to store user account records. The database resides in /var/lib/mysql/mysql.
  - cbgweb—Used by the ClearCount web server to store records for viable HTTPS sessions and the web activity log. The database resides in /var/lib/mysql/cbgweb.
  - Election-specific—One database per election contains its election definition (ballot layout; contest, candidate, and precinct information; and so on), ballot tabulation results, and election activity log.
- CIFS file system server—All ClearCount executable files and configuration files used by the ScanStations reside on the server's file system and are accessed using the common Internet file system (CIFS) protocol. This is implemented on the ScanServer using the commercial off-the-shelf (COTS) Samba software.
- Apache web server—This server receives all incoming HTTPS requests. It is configured to dispatch all URLs to the ClearCount web server.
- ClearCount web server—The server processes and responds to all Clear Ballot HTTP requests. Many of those requests are translated to local transactions with the local MySQL server.

ScanStations

A ScanStation computer has an attached scanner and connects to the ScanServer. The ScanStation computers adjudicate scanned cards and send those results to the ScanServer using Clear Ballot HTTP requests. They also send the card images to the ScanServer using Clear Ballot HTTP requests.

Election administration stations

At least one election administration station is connected to the ScanServer. Using Clear Ballot HTTP requests, the ScanServer sends CSS, JS, and PNG files to election administration stations for creating and managing user accounts; creating and managing election databases; monitoring, generating, and reporting election results; and viewing logs—all via browser pages.

1.1 Applicable documents

Responsive to VVSG 2005, Volume 2, Section 2.5.2.

The following documents control the development of ClearVote software and its specifications:

- The Clear Ballot coding style for Python is controlled by the *PEP 8* standard.
  See https://www.python.org/dev/peps/pep-0008/.
- The Clear Ballot coding style for Javascript is controlled by the *Crockford Style* conventions.
  See http://javascript.crockford.com/code.html.
- Clear Ballot uses GitLab as its source code version control system.
  See https://about.gitlab.com/product/.
- Clear Ballot uses MadCap Flare as its primary authoring tool for all specifications and documents.
  See http://www.madcapsoftware.com/support/onlinehelp.aspx.

Chapter 2. Security overview

This chapter provides an overview of security concepts relating to:

- User access levels
- MySQL user identification

See the *ClearCount Security Specification* for additional information.

2.1 User access levels

Users' access to MySQL databases is based on their MySQL privileges. To hide the details of MySQL, ClearCount user access levels are used in all user interfaces. These access levels are mapped to a set of MySQL privileges that are stored in the mysql database in various tables. The details of MySQL are hidden to simplify the user experience, and to allow future modifications to the ClearCount system without requiring significant changes to the user interface.

The ClearCount user access levels are:

- useradmin—The user can create user accounts.
- dbadmin—The user can change the database phase, and to create, back up, restore, merge, or delete databases.
- modify—The user can readjudicate a ballot.
- append—The user can append ballot adjudication and log entries to an election database.
- read—The user can only read information stored in the database.
- none—No access is allowed.

2.2 MySQL user identification

The name of a ClearCount user is stored in the mysql database user table (mysql.user). The name listed in this table is always associated with the name of a host. The host name identifies the host system that the named user is allowed to use to access MySQL. The host system identity can be an Internet protocol (IP) address, or a name that is mapped to an IP address. The user name and host combination can be represented as: *userName@hostName*.

The host for all ClearCount user accounts is *localhost*, which means that the account is not usable for remote network access. Instead, the user makes page requests from a browser, then the web server processes the URL and makes HTTP message requests on the user's behalf by specifying the user name, password, and host name (*localhost*) during that access.

The Tabulator instance running on a ScanStation is an exception in that it makes HTTP message requests directly. The user name is always *ScanStation*. The MySQL account set up for the ScanStation user sets the host name to %. MySQL interprets host name *%* to mean *any host*. Therefore, the ScanStation is recognized running on any host.

As described in the *ClearCount Security Specification*, the election official must supply the correct password when starting the Tabulator application, and the ScanStation account only has the ability to append log entries and ballot adjudication results to the election database if it is in the Scanning phase and the election is the active election.

2.3 Election database phases

All of the information for a given election, including the ballot layout, contests, candidates, and ballot tabulation results, is contained in a single MySQL database referred to as an *election database*. Each database is assigned a phase that, as expressed in the ClearCount user interface, is one of:

- Scanning—ScanStations can append adjudication results to the database. Ballot resolution officials (any official with modify access level to the database) can manually readjudicate a ballot.
- Reviewing—The ballot scanning process is complete and the results are subject to review by the election personnel. Card images and adjudication results sent by a ScanStation are not accepted by the ScanServer. Ballot resolution officials (any official with modify access level to the database) can manually readjudicate a ballot.
- Closed—The scanning and readjudication processes are complete and the election database can no longer be modified. Election officials can access reports from the database and publish the results.

2.3.1 Election database lifecycle

The concept behind an election database phase is related to the election database lifecycle—how it is created and how it is used. This lifecycle proceeds as follows:

1. The jurisdiction creates a set of PDF files that contain descriptions of ballots. Each file contains one or more PDF pages, and each page describes one side of a ballot.
2. The PDF files can be sent to a printer to produce paper copies of unvoted ballots to be used to conduct the election. These same PDF files are sent to Clear Ballot for processing.
3. Clear Ballot analyzes the PDF files and produces a set of comma-separated values (CSV) files, called a *ballot definition file (BDF)*. Clear Ballot creates a single zipped BDF file and sends it to the jurisdiction.

   Jurisdictions can create their own BDFs in the ClearDesign election management system (EMS). See the *ClearDesign User Guide* and the *ClearDesign Ballot Definition File Guide* for details.
4. The jurisdiction uses the ClearCount Election Administration pages to create a MySQL election database from the zipped BDF file. Each BDF CSV file is converted to an election database table with the same name as the file, but without the .csv extension. This set of BDF tables contains all the information known about the election before a single ballot is scanned. After the tables are created, they are never modified.

5. The Election Administration pages add database tables for collecting ballot tabulation results. These tables are empty when they are created. Entries are added to each table as the cards are scanned to count the votes.

6. The Election Administration pages select the election database as the active database and sets its phase to Scanning. The database is now ready to accept the tabulation results from scanned ballot cards.

7. The ScanStation runs the CurrentElection.bat file, which resides in a shared folder on the ScanServer (and which has been mapped to a local drive letter on the ScanStation) to set local environment variables for the names of the ScanServer and the active election.

8. A Tabulator instance runs on the ScanStation. The Tabulator application uses the environment variables to discover the names of the active election and the ScanServer. The ScanServer runs both the MySQL server and the web server used to process Clear Ballot HTTP requests.

9. The Tabulator application uses Clear Ballot HTTP requests to fetch the BDF tables from the active election database.

10. Cards are scanned at the ScanStation, and are stored in the local file system by the scanner software as JPEG images. The Tabulator application finds and analyzes the JPEG card image files, and then sends the card images and the adjudication results to the ScanServer using Clear Ballot HTTP requests.

11. When all of the ballots have been scanned, an election official sets the election database phase to Reviewing. The ScanServer then no longer allows the ScanStations to add adjudication results to the database, but allows other election officials (with a sufficient access level) to fix problems in the tabulation results using the Card Resolutions tool on an election administration station.

12. After all tabulation problems have been resolved, an election official sets the database phase to Closed. The database is now read-only, but can be accessed to generate reports for the election results.

Access to each database depends in part upon the type of access being requested, the user's access level, and the phase of the election's central-count process.

2.4 Artifacts

In the most general sense, an artifact is any object created by the software. Artifacts can be stored either in-memory or on-disk.

An example of an in-memory artifact is the set of ballot definition tables downloaded by the ScanStation Tabulator application from the active election database on the ScanServer.

In-memory artifacts are not addressed in this document. All important artifacts are written to disk, and any artifact that is stored only in memory is temporary and is recreated if any relevant component of the system is stopped and restarted.

2.4.1 ScanStation artifacts

2.4.1.1 Card images

The only on-disk artifacts stored on a ScanStation are the files containing voted card images and target card images. It is assumed that there are no security issues with respect to accessing these artifacts because the ScanStation does not allow incoming network connections, all outgoing connections are made over a private physical (not Wi-Fi) encrypted Ethernet, and the ScanStation runs only trusted software.

The card images are created by the COTS scanner software, and can be one of several file formats (such as TIFF, JPEG, PNG). The ScanStation sends the card image files to the ScanServer for storage and then deletes the local copy. The filename is scanner-dependent. For example, the Fujitsu scanner software uses the format *<boxID>+<ballotImageSequenceNo>.jpg* where:

- *boxID* is encoded on the target card that precedes a batch of ballots and identifies the batch. It uses the format *counterGroupID-cgSequenceNo* where:
    - *counterGroupID* is an alphanumeric (typically two uppercase alphabetic characters) label associated with the counter group for this batch of ballots. For example: AB for absentee ballot, EV for early voting, and ED for Election Day.
    - *cgSequenceNo* is a sequence number encoded on the target card. No two target cards use the same *counterGroupID* and *cgSequenceNo* combination of values.
- *ballotImageSequenceNo* is a five-digit card image sequence number beginning with 10001. Each scanned card has two sides, and each side is assigned a consecutive image sequence number.

2.4.2 ScanServer artifacts

The ScanServer manages the following artifacts:

- mysql database
- cbgweb database
- web activity log
- election databases
- election activity logs
- files containing the voted card images

2.4.2.1 mysql database

The mysql database is created during server installation. The database contains account entries for all users, along with their passwords, MySQL privileges (and the names of the databases to which those privileges apply), and the remote name (or IP address) of the host the user is allowed to use to make the access request.

Entries in the mysql.user table are always located by a combination of the user name and the host identifier. Entries in the mysql.db table are always located by a combination of the database name, the user name, and the host identifier.

MySQL checks the privileges specified in the mysql database on all access requests to any databases to determine if the accessing user has the privilege to perform the requested access.

An access request can be made:

- Remotely—Using a Clear Ballot HTTP request initiated by a ScanStation (remote) client. The name of the user is specified in the request. The host name comes from the remote IP address.
- Locally—By the web server, on behalf of a ScanStation client, in response to incoming Clear Ballot HTTP requests. The web server specifies the user name and password of the ScanStation user in the local HTTP request. The host name is *localhost*.

2.4.2.2 cbgweb database

The cbgweb database is created during server installation. It contains:

- The cbgweb.log table used to store the web activity log.
- The cbgweb.sessions table. Each record in this table contains information about each viable incoming session, including a session-ID (used in client cookies to identify the session), the remote user name, and the remote user password. The following table is flushed upon web server startup, and is updated as remote users access the server.

2.4.2.3 Web activity log

The cbgweb.log table is used to record all nonelection-specific activity. A log record is appended when users are added or removed, or when their access levels are modified. A log record is also appended when elections are created or destroyed. See the web activity log description in "ClearCount log files" in the *ClearCount Election Administration Guide*.

2.4.2.4 Election database

An election database is created for each election during election setup. Each database contains an election-specific activity log and tables that define the election, and an election definition (using the BDF tables). The metadata table in this database specifies the current phase of the database (Scanning, Reviewing, Closed). Other tables record the election tabulation results.

2.4.2.5 Election activity log

An election-specific activity log is the log table in an election database. It is created during election setup, and is updated whenever there is any election-related activity. See the election activity log description in "ClearCount log files" in the *ClearCount Election Administration Guide*.

2.4.2.6 Voted card images on the ScanServer

The image files of all voted ballots (and target cards) are created and named when the ballot cards are originally scanned on the ScanStation. They are sent to the ScanServer, which stores them as files in a folder (this folder has the same name as the election). There are two files per card—one each for the front and back of the card.

Chapter 3. Software overview

Responsive to VVSG 2005, Volume 2, Section 2.5.3.

The ClearCount system comprises a network of computers that includes:

- One ScanServer that collects tabulation results and runs the software for managing the election
- At least one ScanStation (with an attached scanner) for scanning ballots, capturing voter marks, and sending card images and mark information to the ScanServer
- At least one election administration station for managing the election, adjudicating unreadable cards, and reporting election results

Clear Ballot attests that all procured software is obtained directly from the manufacturer or a licensed dealer or distributor.

3.1 ScanServer

There is one ScanServer in the network. It runs the Ubuntu implementation of the Linux operating system. Applications running on this server include:

- CIFS
- MySQL
- Apache web server
- ClearCount web server

3.1.1 CIFS

The Samba daemon implements the CIFS (common Internet file system) protocol. CIFS is used by the ScanServer to download the ClearCount application executable files. The ScanServer Samba daemon is reachable on TCP port 445. For historical reasons, CIFS is sometimes referred to by the original Windows name SMB (hence the name Samba), which stands for server message block.

There is no DNS (domain name service) server running on the Ubuntu Linux server, the network switch, or anywhere else. The name (and address) of the server is known to clients only because the Samba server provides support for the WINS (Windows Internet Name Service) protocol, which provides translation of SMB share host/folder names to addresses.

3.1.2 MySQL

The MySQL database server process manages databases used to store information about the election (ballot formats, precincts, contests, and so on), ClearCount users, current network sessions, activity logs, and ballot tabulation results. The following databases exist in the ClearCount system:

- MySQL uses a single mysql database, primarily to store user privileges.
- The ClearCount system uses a single cbgweb database to store information about network sessions and to house the web activity log.
- One database per election stores the election definition, the tabulation results, and the election-specific election activity log.

The MySQL process is reachable on TCP port 3306 using MySQL database transactions. The only databases accessed over this port are the election databases, and it is only the ScanServer that uses network MySQL database transactions to access those databases.

Also, local (nonnetwork) MySQL transactions are used by the ClearCount web server to access all databases. This is usually done while servicing an incoming HTTPS request from a client (that is, a browser page running on an election administration station).

3.1.3 Apache web server

The Apache web (HTTPS) server process is reachable on TCP port 80. Apache is configured to dispatch incoming HTTPS messages to a process based on the URL specified in the HTTPS request.

3.1.4 ClearCount web server

The ClearCount web server process is dispatched by Apache. Apache implements all Clear Ballot HTTP requests. For convenience, within this document, the term *web server* refers to the ClearCount web server process and not the Apache server process (because Apache acts merely as a front-end to dispatch to the ClearCount web server).

The IT/system administrator role has direct access to the ScanServer. No other personnel are associated with the ScanServer. All election-related tasks are performed via either MySQL or transactions using Clear Ballot HTTP requests

3.1.5 ScanServer directory structure

ClearCount executable files are located on the ScanServer in the directories listed on the About this Software (/about) page. The complete list of folders containing ClearCount-related files follows:

- /usr/share/cbg/web—Contains Linux executable files and read-only data files on the ScanServer.
- /usr/share/cbg/client—Contains Windows executable files and read-only data files on ScanStations.
- /etc/cbg—Contains ClearCount configuration files.
- /var/cbg/Ballots—Contains card image files for all elections.

The following configuration files are supplied or modified for ClearCount-specific configuration parameters in Linux third-party packages required by the ClearCount system:

- /etc/apache2/conf.d/cbgweb—Configuration settings for Apache Web Server
- /etc/mysql/conf.d/cbg.cnf—Configuration settings for MySQL Database Server

- /etc/samba/smb.conf—Configuration settings for Samba File Server
- /etc/rsyslog.d/cbg.conf—Configuration settings for Linux rsyslog log server
- /etc/init/BallotMonitor.conf—Configuration settings for Linux init server

3.2 ScanStation

The ScanStation scans and adjudicates ballots and sends the card images and the adjudication results to the ScanServer. Each station consists of a computer running Microsoft Windows with a Fujitsu scanner attached to it.

An election official administrator with knowledge of the ScanStation account password turns on the ScanStation and the scanner. This administrator ensures that the scanner is ready to scan cards, and then starts the Tabulator application. Consult the *ClearCount Election Administration Guide* for details about preparing the scanner and starting the Tabulator application.

When the Tabulator process starts, it loads the ballot definitions and other election-specific information from the database via HTTP requests sent to the ScanServer.

A scanner operator then feeds ballots into the scanner. The ScanStation analyzes the ballots and sends the voter mark adjudication results, the card images, and activity logging information to the ScanServer.

The ScanStation uses these protocols to access the ScanServer:

- CIFS:
  - Used to run the CurrentElections.bat file, which sets the environment variables naming the active election and the ScanServer
  - Used to download the Tabulator application
- Clear Ballot HTTP requests:
  - Used to load the election definition file data from the active election's database, which resides on the ScanServer
  - Used to send the adjudication result records, which are appended to the active election database
  - Used to send the card image files. The card image files are stored on the ScanServer in a folder that is exclusive to that election. The folder name matches the election name, and the filename is created at the ScanStation.
  - Used to append entries to an election-specific election activity log.

The following sections describe processes that run on the ScanStation, including:

- COTS scanner software
- The Tabulator application
- The DeleteBox utility

3.2.1 COTS scanner software

The scanner hardware manufacturers provide COTS software that converts scanned card images to output files that are saved to the ScanStation hard drive. For example, the Fujitsiu scanners use the ScandAll PRO software to convert the images to JPEG files.

3.2.2 The Tabulator application

The Tabulator application is a process written by Clear Ballot. It is responsible for finding the card image files stored by the scanner software. It reads and analyzes these image files to discover voter intent, and sends the images and adjudication results to the ScanServer via HTTP message requests.

For a detailed description of how the Tabulator application processes ballots, see "Recording votes accurately" in the *ClearCount Functionality Description*.

3.2.3 The DeleteBox utility

The DeleteBox utility is a process written by Clear Ballot. To delete a box of scanned cards from a ScanStation, the scanner operator shuts down the Tabulator application, and a supervisor launches the DeleteBox utility. The DeleteBox utility allows the supervisor to enter the identity of the box of cards whose results (tabulation results and card image files) are to be removed from the ScanServer. The DeleteBox utility uses the Clear Ballot HTTP requests to communicate with the ScanServer.

For the detailed description of how the DeleteBox utility processes ballots, see "Box deletion requirements" in the *ClearCount Functionality Description*.

3.3 Election administration stations

Election administration stations are computers connected to the ScanServer that run browsers. When browsing to the ScanServer main page (the Election Index page), either by specifying its name or its IP address, a browser page displays to accept the name and password of a ClearCount account. The administrator logs in to the ScanServer web server by entering an account name and password and clicking the **Sign In** button. If the name and password are not recognized by the web server, the login page reappears.

When the account name and password are recognized, the user is logged in and the Election Index page displays. In the upper-right corner of this page, the user name appears, along with a drop-down menu. The contents of this menu depend upon the access level associated with the user, as described in the following table:

Table 3-1. Election administration station menu items

Access level	Action when selected	Used by (role)
Election Administration option		
useradmin dbadmin	Displays the Election Administration page. This page allows the current phase of an election to be changed, and allows the choice of the active election to be changed. Other aspects of election administration include the ability to create, delete, back up, restore, and merge elections.	Principle supervisor Assistant supervisor
User Administration option		
useradmin	Displays the User Administration page. This page allows users to be added and deleted, allows a user's name and password to be changed, and allows a user's access level to be modified.	Principle supervisor
Web Activity Log option		
useradmin dbadmin modify read none	Displays a page that shows the contents of the web activity log. For users with an access level of none, an error is displayed.	Principle supervisor Assistant supervisor Ballot resolution official Recordkeeper Public relations official
Log Out option		
useradmin dbadmin modify read none	The current user is logged out.	

The Election Index page provides a table of election databases to which the accessing user has at least read access (either by default or explicitly for that database). The first column of the table lists the names of the elections. Clicking an election name displays the Dashboard page for that election.

The Dashboard contains links to various reports that can be accessed by the following personnel:

- Priniciple supervisor
- Assistant supervisor
- Scanning supervisor
- Ballot resolution official

- Recordkeeper
- Public relations official

3.3.1 The Dashboard

For more information about the Dashboard, see "Dashboard" in the *ClearCount Reporting Guide*. The Dashboard comprises the following sections.

3.3.1.1 Election data

This section lists elements describing the election, such as the ballot styles, contests, parties, counter groups, and so on. To the right of each element listed, a number appears indicating how many of those elements exist. Clicking one of those numbers displays a page that presents additional details about the element.

3.3.1.2 Ballot-scanning operations

This section lists such data as the number of scanners, the number of boxes scanned, the number of pages (ballot and nonballot cards) scanned, and so on. Clicking the number for a particular item displays a page containing additional information.

3.3.1.3 Visual resolution of unreadable cards

This section provides details of the cards that the ClearCount system is initially unable to adjudicate. Information includes current resolution status data.

See "Resolving card tabulation issues" in the *ClearCount Election Administration Guide* for information about processing unreadable cards.

3.3.1.4 Card reconciliation

This section provides links to reports about the cards themselves, including reports about cards that were unreadable and in need of manual resolution. Clicking the number for cards with this designation displays the Card Images report so the card details can be viewed, and manually resolved, if necessary. (The database does not need to be associated with the currently active election, but the election must be in either the Scanning or Reviewing phase.)

Card resolution can be performed by the following personnel:

- Principle supervisor
- Assistant supervisor
- Ballot resolution official

3.4 Software item identification

3.4.1 ScanStation software

3.4.1.1 ScanStation overview

ScanStation installation includes the following setup:

- The Fujitsu ScandAll PRO application is installed. ScandAll PRO receives the scanned images from the scanner, converts them to an image format, and stores them on a local drive.
- The TWAIN driver is installed. This driver is the software interface between the physical scanner and the ScandAll PRO application.
- A network drive letter is mapped to the client folder on the ScanServer. Using the CIFS protocol, all ScanStation ClearCount software is accessed at run time from this drive.

Election personnel run software on the ScanStation to accomplish the following tasks (all software runs from the mapped client folder shared by the ScanServer):

- Scan and process ballots.
  a. The scanning supervisor runs the appropriate Update*ScannerModel*.bat file for the Fujitsu scanner being used. The *ScannerModel* value corresponds to the scanner model, for example, *Update6800* identifies the Fujitsu fi-6800 scanner. This step is performed at start-of-day and whenever the ScanStation computer restarts.
     i. The Update*ScannerModel*.bat process copies the *ScannerModel*Profile.dsi and *ScannerModel*Profile.ini configuration files locally so the Fujitsu scanner software can find them.
     ii. The Update*ScannerModel*.bat process obtains the scanner model and serial number by running the GetScannerInfo.exe file.
     iii. The Update*ScannerModel*.bat process runs the SetupScanner.exe process so the Fujitsu ScandAll Pro software runs automatically when the operator presses the Scan button on the scanner.
  b. The scanning supervisor then runs the StartTabulator.cmd process, which calls the CurrentElection.bat process to obtain the names of the ScanServer and the active election.
     i. The StartTabulator.cmd process copies the BallotTabulator.exe file (Tabulator application) to a local temporary folder, and then runs that copy of the Tabulator instance. The local copy of the BallotTabulator.exe file ensures its availability if the ScanServer is unexpectedly offline.
     ii. The BallotTabulator.exe process prompts the scanning supervisor for the ScanStation account password.
     iii. The BallotTabulator.exe process continues to run so that ballots can be scanned and tabulated. While running, the BallotTabulator.exe process displays a window to show progress. The BallotTabulator.exe process can be shut down by closing this window.

- Delete from the election database a box of previously scanned and tabulated ballots.
  a. The scanning supervisor runs the DeleteBox.bat file.
  b. The DeleteBox.bat process calls the CurrentElection.bat process to get the names of the ScanServer and the currently active election.
  c. The DeleteBox.bat process runs the BallotDeleteBox.exe file.
  d. The BallotDeleteBox.exe process prompts the scanning supervisor for the identifier of the box of cards to be deleted, and uses an HTTPS request to delete the box of cards from the election database on the ScanServer.
  e. The ScanServer prompts the scanning supervisor for a name and password of a ClearCount account with an access level of modify, and deletes the box of cards.
  f. The BallotDeleteBox.exe process exits.

3.4.1.2 ScanStation software items

The files in the following table are written in-house by Clear Ballot, and contain executable software. The ScanStation either executes the file directly from the ScanServer folder or copies the file to a local folder and executes it there, as described in the table below. Files reside in the /usr/share/cbg/client/ folder on the ScanServer, which is readable over the network using CIFS.

Table 3-2. Executable software files

File	Description	Interface to other items
GetScannerInfo.exe	Executable file run by Update*ScannerModel*.bat to obtain the model and serial numbers of the attached scanner When built, this program links with a third-party Python package named *twain* (see http://sourceforge.net/projects/twainmodule), which reads the scanner hardware model and serial numbers.	The command line is: GetScannerInfo There are no command line arguments.

File	Description	Interface to other items
SetupScanner.exe	Executable file run by Update*ScannerModel*.bat to cause the ScandAll PRO process to be run automatically whenever the scanner's Scan button is pressed	The command line is: SetupScanner.exe *scannerModel scannerProfile* --log=%TEMP%\SetupScanner.log where: *scannerModel* is the scanner model number, for example, 6800 for the fi-6800 model. *scannerProfile* is the root name of the appropriate .dsi and .ini files. For example for the fi-6800 model: CBG fi6800 Profile %TEMP%\SetupScanner.log is the name of a log. This log file is created in the %TEMP% folder (usually C:\tmp or C:\temp). After SetupScanner.exe runs, the Update*ScannerModel*.bat command file types the log file contents and then deletes the log file.
BallotTabulator.exe	Executable file run on a ScanStation to analyze scanned cards and target cards, and to send the automated adjudication results to the ScanServer	Finds a pair of consecutively named card image files in the C:\CBGBallotImages folder and interprets them as the front and back of a card. Uses Clear Ballot HTTP requests to send the card image files and the automated adjudication results to the server.

File	Description	Interface to other items
BallotDeleteBox.exe	Executable file run on a ScanStation to delete an entire batch of previously scanned cards identified by their box ID	Uses Clear Ballot HTTP requests to request that the server delete all ballots associated with a given box ID, both their tabulation results in the election database and the card images stored on the file system.
DeleteBox.bat	Batch file used to launch BallotDeleteBox.exe	Calls CurrentElection.bat to get the names of the ScanServer and the election. Runs BallotDeleteBox.exe.
Update6400.bat	Batch file to set the proper ScandAll PRO profile on a ScanStation, and to obtain scanner model and serial information	Copies \fiG140\CBG fi-6400 Profile .ini and .dsi files to C:\ProgramData\ScandAllPro\public_profiles. Runs GetScannerInfo.exe
Update6800.bat	Batch file to set the proper ScandAll PRO profile on a ScanStation, and to obtain scanner model and serial information	Copies \fiG140\CBG fi-6800 Profile .ini and .dsi files to C:\ProgramData\ScandAllPro\public_profiles. Runs GetScannerInfo.exe
Update6800-18-22.bat	Same as Update6800, but uses a different profile that works better on 18"–22" ballot cards	Copies \fiG140\CBG fi-6800 Profile .ini and .dsi files to C:\ProgramData\ScandAllPro\public_profiles. Runs GetScannerInfo.exe
Update7180.bat	Batch file to set the proper ScandAll PRO profile on a ScanStation, and to obtain scanner model and serial information	Copies \fiG140\CBG fi-7180 Profile .ini and .dsi files to C:\ProgramData\ScandAllPro\public_profiles. Runs GetScannerInfo.exe

File	Description	Interface to other items
StartTabulator.cmd	Command file used to launch BallotTabulator.exe	Calls CurrentElection.bat to get the names of the ScanServer and the election. Runs the Windows taskkill.exe program to stop any currently running Tabulator instance. Copies BallotTabulator.exe to a temp directory and runs it from there.

3.4.1.3 ScanStation configuration data files

The files in the following table are written in-house by Clear Ballot, and contain configuration data for the scanner subsystem. They are copied as needed by the ScanStation to its local folder at C:\ProgramData\ScandAllPro\public_profiles, where they are found by the ScandAll PRO program (written by Fujitsu).

They reside in the /usr/share/cbg/client/ folder on the ScanServer, which is readable over the network using CIFS.

Table 3-3. Configuration data files

File	Description
fi6400/CBG fi6400 Profile.dsi	ScandAll PRO profile for fi-6400
fi6400/CBG fi6400 Profile.ini	ScandAll PRO profile for fi-6400
fi6800/CBG fi6800 Profile.dsi	ScandAll PRO profile for fi-6800
fi6800/CBG fi6800 Profile.ini	ScandAll PRO profile for fi-6800
fi7180/CBG fi7180rofile.dsi	ScandAll PRO profile for fi-7180
fi7180/CBG fi718Profile.ini	ScandAll PRO profile for fi-7180
fi6800/CBG fi6800 18-22 Profile.dsi	ScandAll PRO profile for fi-6800 for 18"–22" ballot pages
fi6800/CBG fi6800 18-22 Profile.ini	ScandAll PRO profile for fi-6800 for 18"–22" ballot pages
fi6800/CBG fi6800 Profile_BAR.jpg	Barcode reference image for barcode recognition, used by ScandAll PRO profile for fi-6800

3.4.1.4 ScanStation procured files

The files in the following table are procured and not modified. They are installed by the IT/system administrator on the ScanStation as part of the ScanStation computer preparation before any ClearCount software is run on that computer.

Table 3-4. Procured files

File	Description	Interface to other items
ScandAll PRO (Fujitsu scanner software)	Converts incoming card images to JPEG format and writes them to the local file system.	Interfaces to the TWAIN driver to connect to the scanner. Interfaces to BallotTabulator.exe by writing card image files to the C:\CBGBallotImages\ folder
TWAIN driver (from Fujitsu)	Handles the communication between computer software and the scanner. There is a different version of this file for each scanner model.	Interfaces to ScandAll PRO to connect to the scanner

3.4.2 ScanServer software

3.4.2.1 ScanServer software items

ScanServer software items include the following:

File	Description
webCBG.fcgi	This is the ClearCount web application. It also includes the Python interpreter. It does not include all the web media files, which are stored in separate files.
sql\cbgweb.sql	SQL script used by the server installer only
BallotBackground	This is the ClearCount application that performs background jobs.
BallotMonitor	This is the ClearCount application that monitors the Linux operating system log and moves those log entries into the ClearCount log.
BallotAuditLog	This is the ClearCount application that monitors and reports any interactive Linux user activity or file system activity.

3.4.2.2 ScanServer configuration data files

Configuration data files include the following:

- /etc/cbg/*.*—Read-only Clear Ballot files that provide the paths of folders that contain installed software and the folder that contains card images
- /etc/mysql/conf.d/cbg.cnf—Read-only Clear Ballot configuration file that provides settings required for MySql databases
- /etc/rsyslog.d/cbg.conf—Read-only Clear Ballot configuration file that provides settings required for rsyslog, the Linux system logger
- /etc/apache2/conf.d/cbgweb—Read-only Clear Ballot configuration file that provides settings required for the Apache web server
- /etc/samba/cbg.conf—Read-only Clear Ballot configuration file that provides settings required for the Samba file server

All of these files are created by the ClearCount installer and are *never* modified.

3.4.2.3 ScanServer procured components

The components in the following table are procured and not modified. They are installed by the IT/system administrator on the ScanServer as part of the ScanServer computer preparation before any ClearCount software is run on that computer.

Component
Ubuntu
Python (part of Ubuntu)
Pillow (part of Ubuntu)
MySQLdb (part of Ubuntu)
PyInstaller
PollyReports
OpenSSL
OpenSSL FIPS Object Module

3.4.3 Election administration station software

3.4.3.1 Overview

Each election administration station presents pages through a browser by sending HTTP requests to the web server component on the ScanServer. The application software executes either on the ScanServer or the election administration station. In the latter case, the necessary application software is sent by the ScanServer to the election administration station using Clear Ballot HTTP requests.

The software resides on the ScanServer in one of the following file formats, and is sent to the election administration station to be executed or interpreted:

- CSS (cascading style sheet) is a plain text file format used for formatting content on browser pages. CSS files are used by browser pages to help keep information in the proper display format. CSS files can help define font, size, color, spacing, border and location of HTML information about a browser page, and are also used to maintain a consistent look throughout multiple pages of a website.

  CSS files aid website programmers by allowing a single file which contains all display settings, as well as simplifying HTML by allowing for multiple page layout design.
- JS files contain JavaScript source code and are used mainly to run client-side JavaScript code on a browser page. The JS file contains all the HTML head and body objects in the tags of a browser page.
- PNG is a raster graphics file format used for storing graphical images.

3.4.3.2 Election administration station software items

The files in the following table are written in-house by Clear Ballot, and are sent by the ScanServer to a web browser as needed using Clear Ballot HTTP requests. They are used by the election administration station browser, but reside in the /usr/shared/cbg/web/media/js/ folder on the ScanServer.

Table 3-5. JavaScript files

File	Description
ballot_remaking.js	Ballot Remaking Tool JavaScript
jquery.dataTables.customizations.js	Clear Ballot data table JavaScript
jquery.dataTables.five_button.js	Clear Ballot data table JavaScript

The files in the following table are written in-house by Clear Ballot, and are sent by the ScanServer to a web browser as needed using the Clear Ballot HTTP requests. They are used by the election administration station browser, but reside in the /usr/shared/cbg/web/media/css/ folder on the ScanServer.

Table 3-6. CSS files

File	Description
ballot_remaking.css	Ballot Remaking Tool CSS
style.css	Clear Ballot CSS

The files in the following table are written in-house by Clear Ballot, and are sent by the server to a web browser as needed using Clear Ballot HTTP requests. They are used by the election administration station browser, but reside in the /usr/shared/cbg/web/media/img/ folder on the ScanServer.

Table 3-7. Image files

File
xs-CB-Lockup-Color-RGB-01.png
xs-CB-Logo-White-01.png
xs-CB-Logotype-White-01.png
xs-CB-Lockup-Color-RGB-01.png
favicon.ico

3.4.3.3 Election administration station configuration data files

The ClearCount election administration station computers do not use any configuration data files.

3.4.3.4 Election administration station procured files

The files in the following table are procured and not modified, and are sent by the server to a web browser as needed using Clear Ballot HTTP requests. They reside on the ScanServer in the /usr/shared/cbg/web/media/img/ folder .

Table 3-8. Procured image files

File
back_disabled.png
back_enabled.png
back_enabled_hover.png
ballot_remaking.png

File
copy.png
copy_hover.png
csv.png
csv_hover.png
forward_disabled.png
forward_enabled.png
forward_enabled_hover.png
glyphicons-halflings-white.png
glyphicons-halflings.png
pdf.png
pdf_hover.png
print.png
print_hover.png
sort_asc.png
sort_asc_disabled.png
sort_both.png
sort_desc.png
sort_desc_disabled.png
xls.png
xls_hover.png

The files in the following table are procured and not modified, and are sent by the server to a web browser as needed using Clear Ballot HTTP requests. They are used by the election administration station browser, but reside in the /usr/shared/cbg/web/media/css/ folder on the ScanServer.

Table 3-9. Procured open source files

File	Description
bootstrap.css	Open source "Twitter Bootstrap" CSS
chosen.css	Open source "Chosen" CSS
TableTools.css	Open source "TableTools" CSS
chosen-sprite.png	Open source "Chosen" images
chosen-sprite@2x.png	Open source "Chosen" images

The files in the following table are procured and not modified, and are sent by the server to a web browser as needed using Clear Ballot HTTP requests. They are used by the election administration station browser, but reside in the /usr/shared/cbg/web/media/js/ folder on the ScanServer.

Table 3-10. Procured open source JavaScript files

File	Description
JavaScript "Bootstrap" library bootstrap.js	Open source "Twitter Bootstrap" JavaScript
JavaScript "Chosen" library chosen.jquery.min.js	Open source "Chosen" JavaScript
JavaScript "jQuery" library jquery-1.10.2.min.js	Open source "jQuery" JavaScript
JavaScript "jQuery-migrate" library jquery-migrate-1.2.1.js	Open source "jQuery" JavaScript
JavaScript "DataTables" library jquery.dataTables.min.js	Open source "DataTables" JavaScript
ColVis colVis.js	Open source "ColVis" JavaScript
JavaScript "TableTools" library TableTools.min.js	Open source "TableTools" JavaScript

File	Description
ZeroClipboard ZeroClipboard.js	Open source "ZeroClipboard" JavaScript
JavaScript "FixedHeader" library jquery.FixedHeader.min.js	Open source "FixedHeader" JavaScript
JavaScript "hotkeys" library jquery.hotkeys.js	Open source "Hotkeys" JavaScript
JavaScript "tooltip" library jquery.tooltip.min.js	Open source "Tooltip" JavaScript
JavaScript "pep" library jquery.pep.min.js	Open source "PEP" JavaScript
JavaScript "LESS" library less-1.3.3.min.js	Open source "LESS" JavaScript

The files in the following table are procured and not modified, and are sent by the server to a web browser as needed using Clear Ballot HTTP requests. They reside in the /usr/shared/cbg/media/swf folder.

Table 3-11. Procured Flash file

File	Description
copy_csv_xls_pdf.swf	Open source "ZeroClipboard" Flash

Chapter 4. Software standards and conventions

Responsive to VVSG 2005, Volume 2, Section 2.5.4.

4.1 Software development methodology

Clear Ballot software development is generated through the following phases:

- Requirements management
- Design
- Construction
- Testing

4.2 Software design standards

Clear Ballot implements the product using Python, a high-level language interpreted by an execution framework. This gives the benefits of an extremely fast development process, excellent run-time validation and exception reporting, and an extremely small source-code footprint.

The Clear Ballot software design process is iterative, employing feedback by working directly with customers who are willing to provide feature suggestions.

4.3 Software specification standards

Software specifications are created for all proposed major features. Specifications must contain the feature objectives, list the reviewers, and contain revision history. Specifications must be kept up-to-date throughout the feature development process.

4.4 Software coding standards

ClearCount software is written in the Python language, with very small amounts of JavaScript used on some browser pages.

Clear Ballot writes its Python source code to conform to the industry-standard PEP-8 coding style, which is documented on web sites such as http://legacy.python.org/dev/peps/pep-0008/. PEP-8 is not specific to the voting industry; most Python applications employ this coding style.

Clear Ballot writes its JavaScript source code to conform to Crockford Style industry-standard coding conventions, which is documented on web sites such as http://javascript.crockford.com/code.html. This widely used coding style was written by the creator of the popular JSLint coding style verification tool.

Because Clear Ballot uses industry-standard coding styles for both Python and JavaScript (per *VVSG Volume 2:5.2.6* and EAC RFI 2010-02) the following *VVSG* requirements are not applicable:

- Volume 1:5.2.3.d, 5.2.5, 5.2.7.b-e
- Volume 2:5.4.2

In addition, per *VVSG Volume 2:5.2.6* and EAC RFI 2010-02, source code conforms to the following additional coding style requirements, which are not exempted by use of an industry-standard coding style:

- Volume 1:5.2.7.a (all units must have block comments, with description, inputs, outputs, globals, units called, and so on)
- Volume 1:5.2.3.a through 5.2.3.c
- Volume 1:5.2.3.e through 5.2.3.f
- Volume 1:5.2.4.a
- Volume 2:5.4.1

4.5 Testing and verification standards

Clear Ballot uses the following methodology in validating its code correctness:

- PyLint—A Python coding style verification tool.
  This tool is run over all Python sources periodically, and run over each source module as it is checked in.
- JSHint—A JavaScript coding style verification tool.
  This tool is run over all JavaScript sources periodically, and run over each source module as it is checked in.

4.6 Quality assurance standards

Frequent execution of a quality assurance regression test suite occurs throughout the development process. See the *ClearVote Test and Verification Specification* for details of the pass/fail criteria for each test case. Overall, internal builds and formal releases, whether minor or major, are not released without all test cases passing.

Chapter 5. Software operating environment

Responsive to VVSG 2005, Volume 2, Section 2.5.5.

5.1 Hardware environment and constraints

Responsive to VVSG 2005, Volume 2, Section 2.5.5.1.

Clear Ballot designs its software and then seeks COTS hardware that provides required characteristics, such as parallel processing and sufficient RAM and disk space. As such, the logic and arithmetic capability of the processor, memory read/write characteristics, external memory device characteristics, peripheral device interface hardware, data input/output device protocols, operator controls, indicators, and displays do *not* constrain software design.

For a description of the hardware environment, see the *ClearCount Hardware Specification*.

5.2 Software environment

Responsive to VVSG 2005, Volume 2, Section 2.5.5.2.

The operating systems and all other COTS software for the ScanStation, ScanServer, and election administration station are identified in the *ClearVote Approved Parts List*.

The source file named *package/syslinux/cbg.seed* contains the complete list of all directives used by the operating system DVD installer for the ScanServer. The directives are customized to support Clear Ballot's standalone installation requirements. This file can be examined to discover the installation configuration and the detailed comments explaining the reasoning for each one.

Clear Ballot authors Windows batch (.bat) and command (.cmd) files, Python code, and JavaScript code. Python and JavaScript code are interpreted. PyInstaller is used for compiling the code into a single executable. Clear Ballot also uses third-party tools that are compiled or assembled.

Chapter 6. Software functional specification

Responsive to VVSG 2005, Volume 2, Section 2.5.6.

This section discusses the pre-election, election, and postelection operational modes and functions of the ClearCount system. See "High-level system description" and "Operational environment of the system" in the *ClearCount System Overview* for more information.

6.1 Configurations and operating modes

Responsive to VVSG 2005, Volume 2, Section 2.5.6.1.

6.1.1 Preparing the systems

The network switch, ScanServer, ScanStations, scanners, and election administration stations must be set up as described in the *ClearCount Election Preparation and Installation Guide*.

6.1.2 Setting up for an election

Requirements include:

- ClearCount user accounts
- Ballot PDF page preparation
- Ballot definition file creation
- Election creation
- Central-count location preparation
- Pre-election testing
- Zero report generation

6.1.2.1 Setting up ClearCount accounts

Two ClearCount user accounts are set up as a part of the ScanServer installation. One of them (the ClearCount primary administrator) has the useradmin access level. The name and password for this account are set during initial installation on the ScanServer. The other is for the ScanStation account. The name for this account is always *ScanStation*, and the password is specified during the installation.

Before the ClearCount system is used for an actual election, other accounts are created for election personnel to perform the roles discussed in "User roles" in the *ClearCount Security Specification*. The accounts are created and managed via the User Administration pages as described in "Creating users and elections" in the *ClearCount Election Preparation and Installation Guide*.

The password for any account can be changed by a user with the useradmin access level.

6.1.2.2 Ballot PDF page preparation

This is the first step in preparing an election. The ClearDesign EMS is used to create ballot styles and generate election definitions. Ballot styles are rendered as PDFs. Election definitions are exported to ballot definition files (BDFs) for importing into the ClearCount system.

6.1.2.3 Ballot definition file creation

The BDFs describe everything that is known about an election—the parties, the precincts, the contests, the choices, and an election definition for each ballot style, including the ballot coordinates for the contest and vote target areas. The BDFs enable the Tabulator application and the Card Resolutions tools to parse the card image files generated by the scanner and to adjudicate the results.

Each BDF uses industry-standard CSV format. The first record in the file contains a comma-separated list of the names of fields. Each subsequent record contains a comma-separated list of the values of those fields.

When the election database is created, the BDF database tables are created. Each table is created and populated from a single BDF. The table name is the same as the filename (but without the .csv extension), and database column names are the field names. See "System database" on page 50 for a detailed description of the BDF tables and files.

The complete set of BDFs describing the election are packaged into a single zipped file. This zipped BDF file is one of the required inputs to the election database creation process. It is possible for a jurisdiction to create this zipped BDF file using its own tools and techniques, in which case the sequence outlined below in this section can be skipped. If the jurisdiction decides to engage Clear Ballot to create the zipped BDF file, the following steps are used:

1. Clear Ballot establishes an account and e-mails the jurisdiction a link to its file transfer site.
2. The jurisdiction creates a single zipped file containing one or more PDF files. Each PDF can contain one or more pages. Each page contains an image of one face of a ballot card.
3. This PDF files in this zipped file must in aggregate contain a page for both faces of each and every ballot card used in the election. The pages representing the front and the back of each ballot card must be included and must be next to each other in the sequence of pages in the zipped file.
4. There can be multiple PDF files, but they must all be included in a single zipped file.
5. The jurisdiction uses the file transfer site to send the name of the election and the zipped file to Clear Ballot.
6. Clear Ballot calculates an SHA-256 digest (also known as the *PDF fingerprint*) of the zipped file contents.
7. Clear Ballot analyzes the PDFs to produce a set of BDFs. The metadata BDF includes, among other things, the PDF fingerprint and the date of the most recently dated PDF file. The BDFs are packaged into a single zipped BDF file.

8. Clear Ballot creates a second file containing an SHA-256 digest (also known as the *BDF fingerprint*) of the contents to the zipped BDF file.
9. Clear Ballot e-mails the jurisdiction when the zipped BDF file is ready to retrieve. The e-mail message contains a link to the zipped BDF file, a link to the BDF fingerprint file, and a text string that displays the BDF and PDF fingerprints.
10. The jurisdiction uses the links to download the zipped BDF and fingerprint files.
11. The jurisdiction calculates the PDF fingerprint from the zipped PDF file that it sent earlier. It compares this value against the PDF fingerprint included in the e-mail message sent by Clear Ballot. If they don't match, the BDFs are deemed unacceptable. (For extra security, the jurisdiction can check the PDF fingerprint in the metafile BDF, and this must also match for the BDFs to be deemed acceptable.)
12. The jurisdiction calculates the BDF fingerprint from the contents of the zipped BDF file and compares this to both the BDF fingerprint in the e-mail message and the BDF fingerprint sent in the BDF fingerprint file. If any two of these three fingerprints do not match, the BDFs are deemed unacceptable.
13. If the zipped BDF file is deemed acceptable, the jurisdiction can use it to create the election.

6.1.2.4 Election creation

Each election is created by creating a separate MySQL database for that election and a file system folder used to store the JPEG card image files. The name of the election database is the same as the name of the election, and locates an election database using the election name in a local directory search. The folder (directory) name is also the name of the election.

An election database can be in any one of three phases: Scanning, Reviewing, or Closed. These phases are discussed below. The phase setting is stored in the metadata table. When an election database is created, its initial phase is Scanning. At any time, an election official assigned an access level of dbadmin or higher can change the phase through the Election Administration pages.

See the *ClearCount Election Preparation and Installation Guide* for more information about creating an election.

More than one election can reside on the ScanServer. However, only one election at a time can be selected as the active election. The active election name is specified at end of the ELECTION= line in the CurrentElection.bat file located in the client folder of the ScanServer file system.

Creating an election database automatically makes that election active. An election administration station uses the Election Administration pages to make any election active or nonactive. Whenever a database is made active, because the name of the active database is stored in only one location on the ScanServer, all other databases are automatically made inactive.

The primary administrator creates an election by copying the election's zipped BDF file sent by Clear Ballot to a folder accessible by an election administration station, and uses the Election Administration pages to create the election.

The ClearCount system can create elections from the following BDF formats:

Format	Description	Source
Unsigned, unencrypted *.bdf.zip*	When creating a BDF for a jurisdiction, Clear Ballot uses a secure download utility and an SHA-256 digest to ensure data integrity. Jurisdictions that create their own BDF (using Clear Ballot tools) must ensure the security of the BDF.	Clear Ballot Technical Services, Clear Ballot tools
Signed, unencrypted *.bdf.zip*	Has a digital signature that uses keyed-hash message authentication code (HMAC) to verify data integrity	ClearDesign 1.4 and 1.5 EMS
Encrypted *.bdfx.zip*	Uses the PBKDF2 (Password-Based Key Derivation Function 2) encryption mechanism for added security	ClearDesign 2.0 (and later) EMS

Creating an election includes the following steps:

1. The folder used to store the card image files is created.
2. The election database is created.
    a. The BDFs are imported into correspondingly named database tables.
    b. Additional tables, which are used to collect the tabulation results, are created.
    c. A table is created to contain the election activity log records.
3. The phase is set to Scanning in the election database's metafile table.
4. The new election is made active by:
    - Using MySQL to grant the ScanStation account SELECT and APPEND access to the newly created database (thereby allowing the Tabulator application to use MySQL to append log entries and ballot processing results to it).
    - The ELECTIONS= line in the CurrentElection.bat file is set to ELECTIONS=*electionName*, making the new election the active election.

6.1.2.5 Preparation for opening the polling place

The ClearCount system is a central-count system, therefore, preparing to open the polls is not applicable.

6.1.2.6 Testing prior to opening the central count of an election

The system hardware and software readiness to conduct a central count of an election is described in "ClearCount readiness testing" and "ClearCount logic and accuracy testing" in the *ClearCount Election Preparation and Installation Guide*.

6.1.2.7 Preparing the zero report

After creating the election, but before scanning any ballots, the election administrator creates and examines a *zero report*, which records that all vote totals are set to zero at the start of the election. See "Preparing the zero report" in the *ClearCount Election Preparation and Installation Guide* for more information.

6.1.3 Recording and tabulating cards

The election administrator ensures that the election is active and in the Scanning phase. The scanning supervisor then turns on each ScanStation and starts the Tabulator application on those stations. The ClearCount account name used by the Tabulator application is hardcoded into the software to be *ScanStation*, but the scanning supervisor must supply the account password so the Tabulator application can successfully access the ScanStation.

Scanner operators can then start scanning ballots. Each box of ballots must have an affixed label with its box ID, and the first card in the box must be the target card specifying that identifier.

Scanning a ballot to record card images and tabulation consists of the following steps:

1. The card is scanned by the scanner. The COTS scanner software running on the ScanStation services the scanner port, and converts the raw images sent by the scanner to two JPEG image files for the front and for the back of the card. These images are written to consecutively named files in the C:\CBGBallotImages folder.

2. The Tabulator process (written by Clear Ballot) running on the ScanStation notices that the front and back images have appeared on the file system, and reads and analyzes the files to produce voter mark processing results that it sends to the ScanServer using MySQL, and sends the JPEG images to the ScanServer using Clear Ballot HTTP requests https://*scanserver*/upload.

3. MySQL on the ScanServer updates the election database tables with the tabulation results, and the web server copies the JPEG images to the file system.

6.1.4 Resolving card tabulation problems

There are two broad categories of problems that each require an independent solution. One requires rescanning and retabulating a box of previously scanned ballot cards. The other requires an administrator to digitally retabulate a single previously scanned card without rescanning it.

See "Error recovery" in the *ClearCount Functionality Description* for additional information about the topics discussed in this section.

6.1.4.1 Deleting a box of scanned cards

Occasionally, it is necessary to rescan and retabulate an entire box of cards. Before this can be done, the ballots from that box, which have already been recorded at the ScanServer, must be removed from the record. This is done by identifying each ballot from that box, removing its tabulation results from the election database, and removing its card image file from the ScanServer's file system.

Identifying the ballots from a box of cards is possible because of the unique box ID. This box ID is printed on the label affixed to the box, and is also printed in a barcode on the target card (which is the first card scanned in the box). This box ID is part of the name of every card image file. Also, each ballot card is assigned a card ID. The card ID is the name of the image file that was determined by the Tabulator application to be the front of the card.

The executable file used to delete the box of cards is BallotDeleteBox.exe. This executable file is run by a command file named DeleteBox.bat, which is executed usually from the ScanStation that originally scanned the ballots. However, because that system might have become nonoperational, it is possible to run the DeleteBox utility from an election administration station. The election must be in the Scanning or Reviewing phase.

All of the executable and command files used by the DeleteBox utility reside in a ScanServer folder, which is mapped for network access by the ScanStation using the CIFS protocol.

The DeleteBox utility first determines if the Tabulator application is running on the system, and, if so, exits with a warning that the administrator must first stop the Tabulator application. If the Tabulator application is not running, the DeleteBox utility runs the CurrentElection.bat command file to set the environment variables, which names the active election and the ScanServer. BallotDeleteBox.exe is then run. It prompts for the name and password of an administrator with a minimum access level of modify. It then prompts for the box ID. The DeleteBox utility uses Clear Ballot HTTP requests to delete the box of cards.

6.1.4.2 Resolving a problem with a single card

Sometimes it is necessary to retabulate the results for a single card. This is usually due to the card being marred or damaged such that it cannot be properly scanned. It could also be that after examining the card image (usually because it was overvoted, undervoted, or its vote target scoring was deemed marginal), an election official examining the card did not agree with the current scoring.

A card is retabulated by the Card Resolutions tool, a browser page that allows election officials to review and resolve unreadable cards on any election administration station. The tool is accessed through the election's Dashboard. Election officials can review the unreadable cards that the ClearCount system digitally outstacks, determine whether they are votable, and process them accordingly.

The inputs to the Card Resolutions tool are the name of the election and the card ID. The election name is used to locate the election database and the ScanServer folder that holds the election's card images, and the card ID is used to locate the card's tabulation results and the actual card image file. The output is a change to the ballot tabulation results.

See "Processing unreadable images" in the *ClearCount Functionality Description* for more information.

6.1.5 Closing the polling place

The ClearCount system is a central-count system, therefore, closing the polls is not applicable.

6.1.6 Closing the central count for an election

To close the central count for an election, a user with an access level of dbadmin or higher sets the phase to Closed through the Election Administration page. This action immediately prevents new Tabulator sessions from launching on ScanStations, but allows current Tabulator sessions to run as long as necessary to process ballots, after which they are terminated. The central count is finished as soon as the last open Tabulator session is terminated.

6.1.7 Generating reports

While in the Closed phase, the database cannot be modified for any reason. However, any official logged in with a ClearCount account with a minimum access level of read can view election reports on any election administration station through the election's Dashboard. See URLs that produce tabular reports for the list of report generation URLs.

6.2 Software functions

This section describes, or provides cross-references to, the software's capabilities or methods for detecting or handling:

- Exception conditions
- System failures
- Data input/output errors
- Error logging for audit record generation
- Production of statistical ballot data
- Security monitoring and control

6.2.1 Exception conditions

Exception conditions are handled by the system by classifying the exception as an error and reporting that error to the user through the appropriate pathway. For a specific listing of all errors, see "ClearCount messages" in the *ClearCount Election Administration Guide*.

6.2.2 System failures

The ClearCount system reports failures that cannot be resolved. Reporting occurs in the Tabulator's Alerts pane or on election administration stations. See "ClearCount messages" in the *ClearCount Election Administration Guide* for descriptions of defined error messages.

6.2.3 Data input/output errors

The ClearCount system identifies data input/output errors by determining that corrupted files have transferred or that two or more components of the system are not able to communicate. When this occurs, the system displays errors indicating that communication is currently unavailable or that the specific files were unable to be copied correctly. For a specific listing of all data input/output errors, see "ClearCount messages" in the *ClearCount Election Administration Guide*.

6.2.4 Error logging for audit record generation

There are two classes of activity logs:

- The web activity log is implemented as a table named *log* in the one and only cbgweb database. Each entry in the table represents a single logged event. Entries in this log are, in general, nonelection-specific. See the web activity log description in "ClearCount log files" in the *ClearCount Election Administration Guide*.
- The election activity log is implemented as a table named *log* in the election database for a particular election. Each entry in the table represents a single logged event. Entries in this log are, in general, election-specific. See the election activity log description in "ClearCount log files" in the *ClearCount Election Administration Guide*.

6.2.5 Production of statistical ballot data

The data quality assessment is based on the ability of the system to recognize and tabulate the scanned card images stored in the card image files. This information is tracked by the system and stored in the election database. It is reported on the Dashboard and in the Distinct Causes of Unreadable Cards report. See "Web reports" in the *ClearCount Reporting Guide*.

6.2.6 Security monitoring and control

The security of a ClearCount installation is achieved in two ways. The first is physical. All network connections use a closed, wired Ethernet. Maintaining the physical security of the Ethernet, the network switch, the ScanStations, the scanners, the election administration stations, and the ScanServer is the responsibility of the jurisdiction.

The second way in which security is controlled is through a role-based access control system. See "Security overview" on page 10 for more information.

The ClearCount system tracks all user and system actions (normal use, unexpected events, errors encountered, abnormal) in two logs—a web activity log that covers all of the jurisdiction’s elections, and a separate election activity log for each election.

The ClearCount logs can be viewed on an election administration station by any user with an access level of read or above. The logs can also be printed or exported. An election’s activity log is included in any backup of the election.

The ClearCount logs are read-only. Their contents cannot be modified by any user. Election administrators monitor security by periodically scanning the entries in these logs. Election administrators also monitor security by checking items such as tamper-evident seals and counting room personnel controls.

Chapter 7. Programming specifications

Responsive to VVSG 2005, Volume 2, Section 2.5.7.2.

Clear Ballot specifically selects COTS hardware and third-party software modules that support freedom in the software engineering process. The selected products pose no limitations, constraints, or unusual design requirements upon the ClearCount software.

For example, MySQL server has a table size limit of 4.0 TB and allows up to 4096 columns per table. However, the limitations on table design pose no unusual circumstances for Clear Ballot software engineers because they greatly exceed requirements for the ClearCount system, being unreachable in an election environment. They constrain the end user rather than the software or its development, and are accounted for using typical software flow and data control techniques, such as bounds checking.

Overall operational speed is also considered in software design, resulting in decisions that use currently available, commercial multihyperthreaded CPUs and streamline the processing of image files. These are not limitations, nor do they pose unusual constraints on the software.

7.1 Programming specifications overview

Responsive to VVSG 2005, Volume 2, Sections 2.5.7.1.

This section provides an overview of software design and structure. The ClearCount software is a network-based product as described in "Software overview" on page 16. The major software components are:

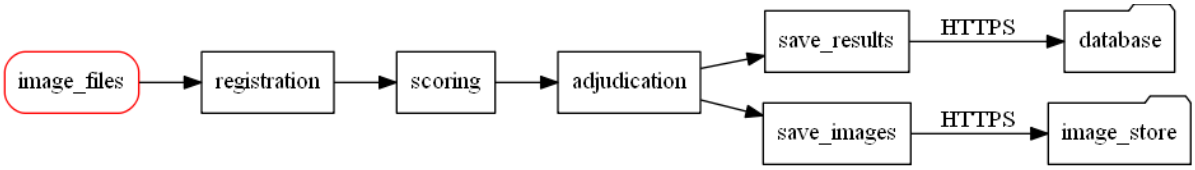
- tabulate folder—Contains the modules that implement the Tabulator application, specifically ballot registration, image analysis, and vote adjudication.
- server folder—Contains the modules that implement the web server applications to perform reporting.
- package folder—Contains build scripts and miscellaneous files necessary to build the product executables.

7.2 Tabulator application (tabulate folder)

The Tabulator application executes on the Windows-based ScanStations, each of which has a single scanner attached to it. The basic workflow is as follows:

1. During initial election setup, the ClearCount software configures the scanner to output image files into a specific folder on the Windows file system called C:\CBGBallotImages. The Tabulator application watches this folder constantly, waiting for newly scanned image files that are generated when the scanner operator scans ballots. Each physical piece of paper is scanned on both sides simultaneously, resulting in two image files for each piece of paper—one for the front and one for the back. As a result, card image files are always processed as pairs.

2. When a previously unprocessed pair of image files appears in the folder, a request to process the images is sent to the Tabulator application.
3. The Tabulator application then performs these actions on the pair of image files:
    a. Registration—Locates the key *landmarks* (different for each type of ballot) that enable code to locate vote targets on the card. The landmarks are used to make the image correspond to the virtual coordinate system used to identify vote target locations.
    b. Scoring—Locates all the vote targets on the cards, based on the landmarks and the virtual coordinates provided by the BDF and stored in the election database tables. After each vote target location is identified, the Tabulator application scores each vote target into a single (floating point) score value, which is used later during vote adjudication. A score value is a floating point value in the range of 0 to 100 that reflects the relative percentage of darkness and coverage of a mark on the vote target. The score values (and associated description of each vote target) for the entire card is combined into one instance of the BallotImageScores object, to be used later.
    c. Adjudication—Analyzes the BallotImageScores object, containing the scores of all the vote targets on both sides of the card, and adjudicates whether each vote target contains a vote or a nonvote, storing the adjudication results into the same object.
    d. Results storage—Stores the results of the card's tabulation by storing new records into the ballots, images, files and scores tables of the database.
4. After processing is performed on a card, a request to save the card images is sent to the Saving component.
5. The Saving component copies both image files to the ScanServer computer over the network, saving the image file content, as well as its name and creation date. After the copy operation is successfully confirmed, the local copies of the image files in the C:\CBGBallotImages folder are deleted.



Tabulator workflow

To achieve parallelism, the Tabulator application launches a number of internal routines as independently operating subprocesses, specifically:

- scanner subprocess—Watches the incoming scanned image folder, and when a new pair of image files appears, it queues a request to the automated adjudication software processes to process this pair of images. It also keeps track of which pairs of image files have been processed to avoid processing any image files more than once.

- tabulation subprocesses—A collection of subprocesses, the number of which is based on the number of CPU hyperthreads, the lowest level of parallelism provided by the computer CPU. For example, on an eight-hyperthread computer, there are seven of these tabulation subprocesses. All of these subprocesses wait for work from a single request queue. When a new work request is placed on the queue, the next available subprocess accepts the work request, and performs the tabulation function as described above.
- saving subprocesses—A collection of subprocesses, the number of which is based on the number of CPU hyperthreads, in the same way as the tabulation subprocesses described above. All of these subprocesses wait for work from a single request queue. When a new work request is placed on the queue, the next available subprocess accepts the work request, and performs the saving function as described above.
- status subprocess—Single subprocess whose sole purpose is to keep the Tabulator window up-to-date with various bits of overall progress information for the scanner operator.

Each of these subprocesses continue to wait for work to perform, until the scanner operator chooses to close the Tabulator application by clicking **Close** on the Tabulator window, at which time all of the above subprocesses exit.

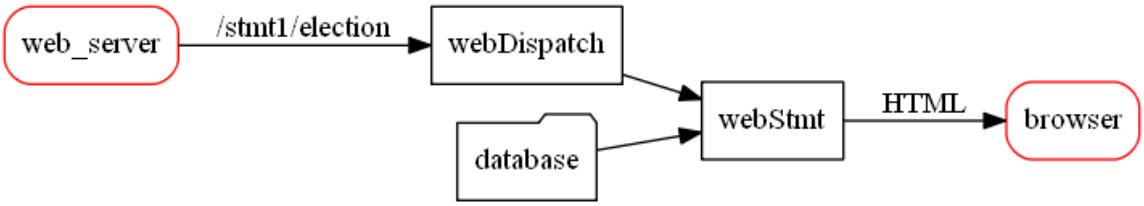
7.3 Server component (server folder)

The server component comprises a web server, running on the ScanServer, which responds to requests from a browser program operated by a human. Requests are made by the browser in the form of HTTP-based web requests, each containing a URL indicating the request to be performed.

The URLs used by this system were designed to be *RESTful*, an industry-standard technique where the URL itself is sufficient to express the request without requiring any internal context stored from previous web requests, and is useful because the URL itself is a full expression of the entirety of the request and all parameters associated with the request. RESTful URLs tend to be very readable by an end user because their format is well-structured, with obvious option names.

When an HTTP request is received by the server, the webDispatch module is called, which parses the URL and passes the request to a web application module. There is one web application module for each type of request. When a web application gets a request, it performs the function and then returns a result back (via HTTPS) to the browser program.

For example, for an HTTP request to return the Statement of Votes Cast report, the web application performs a query of the election database to obtain the necessary information, renders the information as a browser page, and returns the formatted page to the browser client.



Statement of Votes Cast web page generation

7.4 Package component (package folder)

This source folder simply contains build scripts and miscellaneous files related to building product executable files and installation package files.

Chapter 8. System database

Responsive to VVSG 2005, Volume 2, Section 2.5.8.

This section contains an overview of the design and structure of the ClearCount database software.

The MySQL databases used by the ClearCount system include:

- mysql—There is one such database. It is created when MySQL Server is installed, and is used by MySQL Server. The ClearCount web server also uses it for security-related information stored in the mysql.user and mysql.db tables.
- cbgweb—There is one such database. It is created when the ClearCount ScanServer software is installed. It contains these tables:
    - cbgweb.log—Implements the web activity log. See "ClearCount log files" in the *ClearCount Election Administration Guide*.
    - cbgweb.sessions—Contains information about each active network session, including, the last time the session was used, the name of the account being used, and authentication information sent by the user and passed by the web server when issuing MySQL statements on the user's behalf.
- Election databases—There is one such database for each election. The name of the database is the same as the name of the election. It is created when an election is created. See the *ClearCount Database Specification* for details of the contents of an election database.

Additional MySQL artifacts

MySQL artifacts present in MySQL Server include programmic queryable views into the internals of MySQL and are *not* used by the ClearCount system. The views are listed as databases, but cannot store or provide election-related data. However, they must be present for MySQL Server to operate properly. The views include:

- INFORMATION_SCHEMA—Read-only collection of database views for obtaining metadata, such as database name, table names, data types of columns, and access privileges. (The information schema is sometimes called the data dictionary or system catalog.) No files are associated with the views, and programmers cannot set triggers on them.
- PERFORMANCE_SCHEMA—Read-only database view for inspecting internal execution of the server at run time. Code must be specifically implemented for monitoring server performance events. The performance schema uses no persistent storage; the events are not recorded in the server's binary log. The ClearCount code does not implement server performance monitoring.

Chapter 9. Interfaces

Responsive to VVSG 2005, Volume 2, Section 2.5.9.

This chapter describes the interfaces between central-count components described in "Interfaces among internal component" in the *ClearCount System Overview*. These components are:

- ScanServer—A computer that runs the ClearCount software and hosts its election database and the web server that serves its election reports. The ScanServer uses the Ubuntu Linux operating system (a configured version of which is installed with the ClearCount software).
- ScanStations—At least one computer with an attached scanner for scanning and processing voter-marked ballots. ScanStation computers run the Microsoft Windows operating system.
- Election administration stations—At least one Microsoft Windows computer installed with browser software. Election officials use this computer to administer elections and users, to review election reports, and to adjudicate unreadable cards.
- Network switch—Connects the ScanStations and the election administration stations to the ScanServer using a wired, closed Ethernet.

9.1 Interface identification

Responsive to VVSG 2005, Volume 2, Section 2.5.9.

External interfaces

The ClearCount system uses the following external interfaces:

- ScanStation computer to scanner—USB 2.0 or 3.0 cables

  For capturing scanned image files

- ScanStation computer to network switch—wired Ethernet (TCP/IP)

  For sending scanned image files to the ScanServer

- ScanServer computer to network switch—wired Ethernet (TCP/IP)

  For communicating with the ScanStations and the election administration stations

- ScanServer computer to external hard drive—USB 3.0 cable

  For backing up and restoring elections

- Election administration station computer to network switch—wired Ethernet (TCP/IP)

  For communicating with the ScanServer

- Election administration station computer to USB drive—USB 2.0 or 3.0

  For loading BDFs and saving reports

USB interfaces are governed by the USB specification, version 2.0 or 3.0, depending upon the capabilities of both the computer and the connected device. TCP/IP interfaces adhere to the Internet protocol suite of standards.

Internal interfaces

Software components in the ClearCount system interact in either of these ways:

- The software on a computer is seen as layered, with an interface between the upper layer and the layer immediately below it. These interfaces are given in the tables in "Interface description" below. The identifier for each interface is the combination of the first two columns in each table. The first column is the upper layer, and the second column is the layer immediately below the upper layer.
- Between computers, components communicate using a client/server model. The client runs on a ScanStation or an election administration station.

The following interfaces are used:

- HTTP protocol:
  - Used by the ScanStation to obtain the BDFs and to send logged events and ballot tabulation data to the ScanServer. The ScanStation's web server application also uses the HTTP protocol to access databases on behalf of a remote user (a ScanStation or an election administration station) by specifying their account information.
  - Used between a client (a ScanStation or an election administration station) and the ScanServer. The COTS Apache software runs on the ScanServer and forwards all traffic to the ScanServer software written by Clear Ballot. The messages exchanged are defined by Clear Ballot HTTP requests. See "Clear Ballot HTTP requests" on page 68 and "Software item identification" on page 22 for more information.
- CIFS protocol—Allows a client computer to access files in folders that the ScanStation makes available for network access. CIFS is strictly COTS and, therefore, not described in this document.

CIFS and HTTP are industry-standard interfaces. They are not developed by Clear Ballot.

9.2 Interface description

9.2.1 ScanServer interface description

The ScanServer interface includes these components:

- Apache interface—Industry-standard
- ClearCount web server—Developed by Clear Ballot or industry-standard, as noted. Clear Ballot

HTTP requests are closed-network.

- CIFS server—Industry-standard

9.2.2 Apache interface

The Apache interface is a COTS HTTPS server. It is configured to dispatch all incoming URLs to the ClearCount web server.

Table 9-1. Apache interface components

Lower-level component	Interface	Description
LAN (IPv4 over Ethernet)	TCP/IP	This is an internal system interface used by various COTS software components. The ClearCount software never uses this interface directly.
ClearCount web server	fcgid	The interface between the main Apache process and the ClearCount web server uses the industry-standard fcgid protocol, using the standard fcgid interface module.

9.2.3 ClearCount web server

This is an executable written by Clear Ballot to process all incoming URLs and provide election administration stations with its application files, and give them access to the data stored in the MySQL databases.

Table 9-2. ClearCount web server components

Lower-level component	Interface	Description
Developed by Clear Ballot		
Card image files	Clear Ballot HTTP requests	Card images are stored in a separate server folder that is based on the election name. The folder is: /usr/cbg/*electionName* The card images are sent by BallotTabulator.exe using the Clear Ballot HTTP request upload action. They may be later deleted by BallotDeleteBox.exe using the Clear Ballot HTTP request deletebox action.

Lower-level component	Interface	Description
Web application files	Clear Ballot HTTP requests	These are the files sent by the ScanServer to the election administration stations using Clear Ballot HTTP requests.
CurrentElection.bat	Local file system	This command file sets the ELECTION and CBGSERVER environment variables when run on a Windows computer. The CBGSERVER value is set when the ScanServer is installed. The ELECTION value is modified by the web server whenever a new election is made active, or whenever the active election is made inactive.
Industry-standard		
MySQL database	MySQL	MySQL transactions conducted over a local TCP/IP connection. MySQL transactions are initiated with the local MySQL server in response to incoming Clear Ballot HTTP requests sent by an election administration station, or sent by a ScanStation running BallotDeleteBox.exe.

9.2.4 CIFS server components

CIFS server components are COTS software. They manage the server end of the CIFS protocol, which is used by clients to access shared server folders.

Table 9-3. CIFS server components

Lower-level component	Interface	Description
LAN (IPv4 over Ethernet)	TCP/IP	This is an internal system interface used by various COTS software components. The ClearCount software never uses this interface directly.

Lower-level component	Interface	Description
Local files	Local file system	The administrator at a ScanStation or election administration station maps the /usr/share/cbg/client local folder to a network drive. When one of those clients accesses files on that network drive, the CIFS protocol is used, and the CIFS server accesses the file on the folder mentioned above. This is a COTS interface that is not authored by Clear Ballot.

9.2.5 ScanStation interface description

The ScanStation interface includes the following components. The Clear Ballot HTTP requests are closed-network.

Human

A ClearCount administrator.

Table 9-4. Human interface command files

Component	Interface	Description
Update*ScannerModel*.bat	Windows command line	For example: Update6800.bat
ScandAllPro.exe	Windows command line	There are no arguments passed in the command line. ScandAll PRO gets all of its configuration information from the profile .ini and .dsi files in the C:\ProgramData\ScandAllPro\public_profiles folder.
StartTabulator.cmd	Windows command line	This is a Windows command file written by Clear Ballot. There are no parameters.
DeleteBox.bat	Windows command line	DeleteBox.bat *boxID* This is a Windows command file written by Clear Ballot.

Update<xxxx>.bat files

These are command files written by Clear Ballot. They are run by an administrator prior to starting the ScandAll PRO process. See "ScanStation software items" on page 23 for more information.

Table 9-5. ScanStation command files

Component	Interface	Description
copy	Windows shell command	Copies model-specific profile files to the C:\ProgramData\ScandAllPro\public_profiles folder
GetScannerInfo.exe	Windows command line	There are no parameters. All information comes from the profile files that were copied to the C:\ProgramData\ScandAllPro\public_profiles folder.

ScandAllPro.exe

This is COTS software written by Fujitsu that connects to the scanner. It converts incoming scanned card images to JPEG files, which it then writes to the file system.

Table 9-6. ScandAll PRO interface files

Component	Interface	Description
command line	Windows command line	There is no command line. When the process starts, it reads the profile files copied by Update*ScannerModel*.bat to get its configuration information.
configuration information		The ScandAll PRO process gets its configuration information from the following sources: • The profile files (.dsi and .ini) that contain the actual configuration information and are always located in the C:\ProgramData\ScandAllPro\public_profiles folder. • The scanner hardware itself gets additional advanced hardware settings from information stored in the scanner's EPROM. See "Configuring ScanStation computers" in the *ClearCount Election Preparation and Installation Guide* for more information.
scanner driver (TWAIN)	COTS interface	This is COTS software and is not relevant to this document. It is the driver ScandAll PRO uses to access the scanner port.

Component	Interface	Description
BallotTabulator.exe	local file system	As directed by the configuration information, ScandAll PRO converts all scanned images to JPEG files. For each card scanned, there are two images and, therefore, two files, one for each side of a card. These files are copied to the C:\CBGBallotImages folder. The filenames have the following format: *boxID+ballotImageSequenceNo*.jpg • where *boxID* identifies a box or batch of ballots. It is encoded on the target card that precedes a batch of ballots and identifies that box of ballots. • where *ballotImageSequenceNo* is a five-digit card image sequence number beginning with 10001. Each card scanned has two sides, and each side is assigned a consecutive image sequence number by ScandAll PRO. The *boxID* uses the format: *counterGroupID-sequenceNo* • where *counterGroupID* is an alphanumeric (typically two uppercase alphabetic characters) label associated with the counter group for this box of ballots. Examples are: AB for absentee ballot, EV for early voting, ED for Election Day. • where *sequenceNo* is a sequence number encoded on the target card. No two target cards use the same *counterGroupID* and *sequenceNo* combination of values.

StartTabulator.cmd

This command file determines what the active election is, terminates any currently running instances of BallotTabulator.exe, copies a fresh instance of BallotTabulator.exe from the ScanServer to a local file on each connected ScanStation, and runs that local copy.

Table 9-7. StartTabulator.cmd interface components

Component	Interface	Description
CurrentElection.bat	Windows command line	call CurrentElection This is a Windows command file written by Clear Ballot. There are no parameters.
taskkill	Windows shell command	taskkill *params* This is a built-in Windows function.
xcopy	Windows shell command	xcopy BallotTabulator.exe %TEMP% /y /d
BallotTabulator.exe	Windows command line	BallotTabulator.exe *params* This an executable program written by Clear Ballot.

DeleteBox.bat

This command file is written by Clear Ballot. It is run on a ScanStation (or on an election administration station) to launch BallotDeleteBox.exe to delete a box of cards.

Table 9-8. DeleteBox.bat interface components

Component	Interface	Description
CurrentElection.bat	Windows command line	call CurrentElection There are no parameters.
BallotDeleteBox.exe	Windows command line	BallotDeleteBox.exe *params* This an executable program written by Clear Ballot and called by DeleteBox.bat.

BallotTabulator.exe

This is an executable written by Clear Ballot that analyzes the JPEG images of scanned cards, and sends the results and a copy of the JPEG files to the ScanServer.

Table 9-9. BallotTabulator.exe interface components

Component	Interface	Component
command line		The command line is: BallotTabulator *argList* See "Command arguments for BallotTabulator.exe" on page 62.
CurrentElection.bat	Windows command line	call CurrentElection There are no parameters.
ScanServer: getting BDF information		When the Tabulator application gets the BDF information from the ScanServer
Getting local card images from the scanner		The JPEG card image files reside in the C:\CBGBallotImages folder. (See the ScandAll PRO table.) The Tabulator application scans this folder for two JPEG files with names with consecutive sequence numbers, and assumes that these are the images for the front and back of each card scanned. The Tabulator application analyzes the images using the information in the BDF tables to determine which of the two files contains the image for the front of the ballot. The Tabulator application then analyzes each image using the BDF tables to determine the results.
ScanServer: storing tabulation results in database	Clear Ballot HTTP requests	The Tabulator application uses Clear Ballot HTTP requests to store tabulation results in the election database on the ScanServer.
ScanServer: logging events	Clear Ballot HTTP requests	The Tabulator application uses Clear Ballot HTTP requests to log events to the election database on the ScanServer.
ScanServer: storing card images	Clear Ballot HTTP requests	The Tabulator application uses Clear Ballot HTTP requests to send the ballot to the ScanServer.

BallotDeleteBox.exe

This executable is written by Clear Ballot. It is run on a ScanStation or an election administration station to delete a box of cards.

Table 9-10. BallotDeleteBox.exe interface components

Component	Interface	Description
command line		The command line is: BallotDeleteBox *argList boxID* See "Command line arguments for BallotDeleteBox.exe" on page 63.
deleting local card images	local file system	The C:\CBGBallotImages folder is scanned. The ClearCount system deletes all JPEG card image files where the box ID portion of the filename matches the box ID of the box of cards being deleted.
ScanServer: deleting tabulation results	Clear Ballot HTTP requests	BallotDeleteBox uses Clear Ballot HTTP requests to delete boxes of ballots. It specifies the election name and box ID. The ClearCount web server removes all tabulation results for ballots that are associated with this box ID.
ScanServer: deleting card images	Clear Ballot HTTP requests	BallotDeleteBox uses Clear Ballot HTTP requests to delete boxes of ballots. It specifies the election name and box ID. The ClearCount web server removes all JPEG files in the /usr/cbg/*electionName* folder that contain the box ID at the beginning of its name.

CurrentElection.bat

Sets the ELECTION and CBGSERVER local environment variables.

Table 9-11. CurrentElection.bat interface components

Component	Interface	Description
set	Windows shell command	set ELECTION=*electionName* set CBGSERVER=*serverName*
CIFS	CIFS protocol	CurrentElection.bat runs locally on a ScanStation or an election administration station, but resides on the ScanServer in a shared folder as described in the CIFS section of this table.

CIFS

The administrator at a ScanStation maps the /usr/share/cbg/client local folder to a network drive. When one of those clients accesses files on that network drive, the CIFS protocol is used. The network drive mapping and the network interfaces are COTS issues and not ClearCount software issues.

LAN (IPv4 over Ethernet)

This is an internal system interface used by various COTS software components. This interface is never used directly by any ClearCount software.

Windows command shell

Table 9-12. Windows Command Shell interface components

Component	Description
call	Calls another .bat or .cmd file without exiting the command file in which the @call command appears. The only form of this shell command used by the ClearCount command files is: call *fileName* There are no arguments or switches used in the call.
copy	Copies files. For example: copy /y *sourceFilePath destinationFilePath* where */y* tells copy not to prompt to confirm any file overwrites.
echo	Displays a message. The only form of this shell command used by the ClearCount command files is: echo *messageText*
goto	Transfers control to a labeled line within the current command file. goto *labeledLine*
find	Searches for a text string in a file. DeleteBox.bat searches the list of files generated by tasklist to see if the BallotTabulator.exe task is running.
if errorlevel	Specifies a true condition if the last program run returned an exit code equal to or greater than the number specified. DeleteBox.bat uses: if errorlevel 1 goto Start after checking whether the Tabulator application is running.

Component	Description
pause	Suspends processing of a batch program and displays the message *Press any key to continue*.
set	Sets an environment variable. The command is: set *variableName=variableValue*
taskkill	This command is used by StartTabulator.cmd to terminate any currently running instance of a process running BallotTabulator.exe. The command used is: taskkill /F/ FI "IMAGENAME eq BallotTabulator*" where: */F* forcefully terminates the process. */FI* allows the asterisk (*) character to be used as a wildcard specifier.
tasklist	Returns a list of currently running processes. The ClearCount batch files never specify any parameters or switches for this command.
xcopy	Copies files and directory trees. The command used by StartTabulator.cmd assuming the current device and directory is the mapped drive on the server (using CIFS) where the StartTabulator.cmd file resides. The command issued StartTabulator.cmd is: xcopy BallotTabulator.exe %TEMP% /y /d where: *%TEMP%* specifies a drive and directory local to the ScanStation. */y* tells xcopy not to prompt to confirm any file overwrites. */d* tells xcopy to only copy the file if it is newer than an existing copy of the file.

9.2.5.1 Command arguments for BallotTabulator.exe

The basic format of the command is: BallotTabulator.exe arg_1, arg_2, ... , arg_n

The following lists of command line arguments were generated by the command: BallotTabulator.exe -- help

Argument	Description
-h, --help	Show this help message and exit

Argument	Description
-D DATABASE, --database DATABASE	Database name
-H HOST, --host HOST	Database server (default is *localhost*)
-U USER, --user USER	Database user (default is your user name)
-P [PASSWORD], --password [PASSWORD]	Database password (default is a prompt)
-Q PORT, --port PORT	Database server port (default is 3306)
-s SCANNERFOLDER, --scannerfolder SCANNERFOLDER	Input scanner folder (default C:\CBGBallotImages)
-o [OUTPUTFOLDER], --outputfolder [OUTPUTFOLDER]	Optional output folder where images are moved after tabulation (for example, https://ScanServer or X:\)
-p PROCESSES, --processes PROCESSES	Tabulate using multiprocessing (default = # cores - 1)
--savingprocesses PROCESSES	Saving images using multiprocessing (default = # cores - 1)
--scannermodel [MODEL]	Scanner model name (default is obtained from GetScannerInfo)
--scannerserial [SERIAL]	Scanner serial number (default is obtained from GetScannerInfo)
--skiptabulation	If specified, do not attempt to tabulate (only sweep images).
--batch	If specified, treat scanner folder as a noninteractive read-only set of images, and do not delete the images.
--hideconsole	If specified, hide console window (leaving just the GUI window).
--nogui	If specified, skip GUI window (leaving just the console).
-d [LEVEL], --debug [LEVEL]	Trace details of file as it is being read
-R, --reset	Delete and recreate output tables prior to start
-E [SECS], --debugger [SECS]	Allow remote debugging (secs to delay for attach)

9.2.5.2 Command line arguments for BallotDeleteBox.exe

The basic format of the command is: BallotDeleteBox.exe BOXID arg_1, arg_2, ... , arg_n

The following lists of command line arguments were generated by the command: BallotDeleteBox.exe BOXID –help

Argument	Description
BOXID	Box IDs to be deleted
-h, --help	Show this help message and exit.
-s SCANNERFOLDER, --scannerfolder SCANNERFOLDER	Scanner output folder
-o ARCHIVEFOLDER, --archivefolder ARCHIVEFOLDER	Folder where images are moved after being adjudicated
-f, --force	Allows deletion from a computer other than the ScanStation.
-D DATABASE, --database DATABASE	Database name
-H HOST, --host HOST	Database server (default is *localhost*)
-U USER, --user USER	Database user (default is your user name)
-P [PASSWORD], --password [PASSWORD]	Database password (default is a prompt)
-Q PORT, --port PORT	Database server port (default is 3306)
-d [LEVEL], --debug [LEVEL]	Trace details of file as it is being read

9.2.6 Election administration station interface description

The election administration stations use two different programs to connect to a ScanServer:

- A browser—The browser connects to the web server running on the ScanServer. The interface consists of the set of HTTP requests documented in "Clear Ballot HTTP requests" on page 68.
- DeleteBox—The DeleteBox utility (DeleteBox.bat runs BallotDeleteBox.exe) is run to delete a batch of previously scanned cards. Most of the time, the DeleteBox utility is run from the same ScanStation that was used to originally scan the ballots. In addition to deleting the card images and tabulation results from the ScanServer, any lingering card images still residing on the ScanStation are deleted. However, when the ScanStation is not operational, the DeleteBox utility is run on an election administration station. The election administration station interface description for DeleteBox is identical to the description in "ScanStation interface description" on page 55.

9.2.7 ScanServer setup interface

The ScanServer is set up by booting from an installation DVD. Before installing any software, the installation script reformats the ScanServer's hard-drive.

The installation DVD contains all the software necessary to completely configure the server, including the Ubuntu Linux operating system, the Apache web server, the CIFS server, and the ClearCount software for running the ScanServer, the ScanStations, and the election administration stations.

Upon completion of the installation, the only files residing on the ScanServer's hard-drive are the files put there during installation. No additional executable software is ever added. The installation DVD is removed and the ScanServer is restarted. At this point, the ScanServer is completely configured and ready for use.

During installation, the following setup occurs:

1. A Linux root account is set up. Its password is specified during installation. This is an operating system account, and is never used. This account is set up because it is a requirement of the MySQL (COTS) installation.

2. The server is assigned an IPv4 address.

3. The server is assigned a host name.

4. The Apache web server is set up to dispatch all incoming URLs to the ClearCount web server.

5. MySQL is installed and the mysql database is created. This database is a special database used by MySQL (and by the ClearCount software). It contains the user table, which contains the MySQL user account records. It also contains a db table that contains information about each election database.

6. A MySQL root account is created. This is a MySQL account, not an operating system account. The account record resides in the mysql.user database table. This account name is *root*. The account password is specified during installation. This account is required for MySQL setup, but is never used after that. However, the account is granted all MySQL privileges if it is used to access MySQL from the localhost, and, for that reason, the password should be chosen according ClearCount password guidelines.

7. The account for the ClearCount principle supervisor role is created. Its name and password are specified during the installation. This is a MySQL account, not an operating system account. It is granted the MySQL privileges associated with the ClearCount useradmin access level. (See the *ClearCount Security Specification*.) This account is used to create, modify, and delete additional ClearCount accounts.

8. The ClearCount ScanStation account is created. This is the account name used by all ScanStations on all ScanServer MySQL transactions. The name is always *ScanStation*. The password is specified during the installation. This account is not granted any default MySQL privileges. However, when a specific election is the active election and its election database is in the Scanning phase, the ClearCount web server (which is responsible for setting which

election is active and for setting its election database phase) grants MySQL privileges to this account for that particular database so the ScanStation can append log entries and voter mark processing results.

9. The cbgweb MySQL database is created. This database is used by the ClearCount server software. It has tables for keeping track of viable HTTPS sessions and for storing the web activity log.
10. The ClearCount server software is installed.
11. The software and other files to be run or downloaded by the ScanStation using the CIFS protocol are copied to the server's hard drive.
12. The ClearCount files to be sent to the election administration stations are copied to the server's hard drive.

9.2.8 Network switch interface

The network switch serves as a COTS appliance used to connect computers via TCP/IP connections. There is no configuration for this appliance. All computers connect to the network switch using wired Ethernet cables.

9.2.9 Election administration station-to-ScanServer interface for creating an election

This interface is used by an election official to create a new election database. From an election administration station, an administrator uses the ClearCount Election Administration page to create the election using Clear Ballot HTTP requests. See "Creating users and elections" in the *ClearCount Election Preparation and Installation Guide* for more information.

The inputs are the name of the election and the path to a zipped BDF file. Each BDF is in industry-standard CSV format, and corresponds to an election database MySQL table. The first record of each file contains the list of names for the fields (column names) for the table. Each subsequent record contains the list of values for a record (row) in the table.

The jurisdiction can create its own zipped BDF file or provide Clear Ballot with a set of PDF files containing pages to be used to print the ballots for an election. Clear Ballot examines the PDFs to produce a set of BDFs. See the *ClearDesign User Guide* and the *ClearDesign Ballot Definition File Guide* for more information.

The output consists of two artifacts created on the ScanServer:

- Election database—Contains tables used to hold the election activity log, a copy of each BDF (and using the same name as each file, but without the .csv extension), and the ballot tabulation results.
- Folder—Used for storing the JPEG images of scanned cards after they have been tabulated. The database name and the name of the folder are the same as the name of the election.

9.2.10 ScanStation-to-ScanServer interface for ballot scanning

These are all network remote interfaces expressed using a network protocol, as follows:

- During ScanServer setup, a drive letter (usually P:) is mapped to the ScanServer client folder. (For example, net use P: \\ScanServer\client.) The CIFS protocol is used to access files on the mapped drive.
- The scanner operator runs the Update*ScannerModel*.bat command file applicable to the scanner being used. (For example, Update6800.bat for the Fujitsu 6800 scanner.) This file copies the .ini and .dsi files locally so the Fujitsu software can find them. It then runs P:\GetScannerInfo.exe to get the scanner model and serial numbers.
- Executing P\:CurrentElections.bat sets the ELECTION environment variable to the name of the current election (and the name of the MySQL election database), and sets the environment variable CBGSERVER to the name of the ScanServer.
- The scanning supervisor runs P:\StartTabulator.cmd. Executing this file runs P:\CurrentElections.bat and then shuts down any currently running Tabulator process, copies the executable file P:\BallotTabulator.exe locally (so that it is available if the ScanServer shuts down), and runs the local copy of BallotTabluator.exe (also known as the Tabulator application).
- The Tabulator application prompts for the password for the ClearCount account named *ScanStation*.
- The Tabulator application uses MySQL to load the BDF tables from the election database. This information is needed to analyze the card images and to generate the voter mark processing results.
- As ballots are fed into the scanner, the Fujitsu software (ScandAll PRO) copies JPEG images of the ballots to the ScanStation's C:\CBGBallotImages local folder.
- The Tabulator scans C:\CBGBallotImages for two consecutively named JPEG files and interprets them as the front and back of a ballot card. It analyzes the images, sends the voter mark processing results to the ScanServer using MySQL, and sends the JPEG images to the ScanServer using HTTPS.

9.2.11 ScanStation-to-ScanServer interface for deleting tabulation results

Sometimes it is necessary to delete tabulation results for an entire box of scanned ballots. This is done by running the DeleteBox utility on the same ScanStation where the box of cards was scanned. By running the DeleteBox utility on the original ScanStation, card images containing the box ID as part of its name residing in the C:\CBGBallotImages are deleted. If the ScanStation is no longer functioning, the DeleteBox utility can be run from an election administration station.

The entire DeleteBox sequence is as follows:

1. During ScanServer setup, a drive letter, usually P:, is mapped to the ScanServer client folder. (For example: net use P: \\ScanServer\client.) The CIFS protocol is used to access files on the

mapped drive.

2. The scanning supervisor runs P:\DeleteBox.bat. This command file first checks to see if the Tabulator application is running, and, if so, displays a message saying that the Tabulator application must be stopped before the DeleteBox utility can be run, and then exits. If the Tabulator application is not currently running, this command file calls P:\CurrentElection.bat and then runs P:\BallotDeleteBox.exe.

3. The DeleteBox utility prompts for the user name and password of a ClearCount administrator. This administrator must be assigned an access level of modify or higher (scanning supervisor, assistant supervisor, principle supervisor). Otherwise, the DeleteBox utility is not allowed to delete any ballots. See "ClearCount user access levels" in the *ClearCount Security Specification* for more information.

4. The DeleteBox utility prompts for the box ID of the ballots to be deleted.

5. The DeleteBox utility uses the "Clear Ballot HTTP requests" below to delete the box of cards. The card ID for each ballot contains the box ID. For each card ID with this box ID, the ScanServer deletes associated tabulation results and card image files.

9.2.12 Election administration station-to-ScanServer Card Resolutions interface

Resolving unreadable cards is a complex, multistep, and interactive task carried out by an election official assigned the role of ballot resolution official using the Dashboard to locate, display, and digitally adjudicate cards that were not properly adjudicated by the Tabulator application. All of the steps necessary to locate, display, remove a card's tabulation results, and retabulate the card are performed by the official using the Dashboard. All of the interactions between the Dashboard and the ScanServer use the "Clear Ballot HTTP requests" below.

For a detailed description of the Dashboard and its Card Resolutions tool, see the *ClearCount Election Administration Guide* and the *ClearCount Reporting Guide*.

9.2.13 Clear Ballot HTTP requests

The election administration stations communicate with the ScanServer web server using Clear Ballot HTTP requests. Except where noted, the HTTP requests are sent by the browser using either the Election Administration or User Administration pages.

The HTTP requests recognized by the ScanServer have the basic form:

https: //*serverName*/*actionType*/*parameterList*

where:

serverName is the name or the IP address of the ScanServer.

actionType specifies the action to be performed.

parameterList is a backslash-separated ordered list of parameters.

For example, sending the server the HTTP request https://Server01/precincts/ma_boston_2018g causes the server to respond with the browser page containing the report for all precincts for election ma_boston_2018g (Boston, Massachusetts, 2018 general election).

The tables that follow describe the HTTP requests recognized by the server. The parameters column in these tables are described as follows:

NA—Not applicable. There are no parameters.

electionName—Name of an election.

contestID—Number associated with a contest. This number shows up in the first column of the Contests report.

CardID—Standard card identifier, for example, AB-001+10003

imageFileName—Standard card image filename, for example, AB-001+10003.jpg

This section describes the message sent by an election administration station or ScanStation.

9.2.13.1 URLs to log in/log out

Action	Parameters	Description
login	NA	Login prompt for ClearCount user name and password
logout	NA	Causes the user to be logged out.

9.2.13.2 URLs that produce static files

Action	Parameters	Description
media	*filePath*	Handles requests for static files

9.2.13.3 URLs that produce tabular reports

Action	Parameters	Item
elections	NA	Election Index
databases	NA	Election Index
dash	NA	Dashboard
indexes	*electionName*	Election Performance Indexes
boxes	*electionName*	Card inventory report

Action	Parameters	Item
precincts	*electionName*	Precinct report
scanstations	*electionName*	ScanStation report
errortypes	*electionName*	Distinct Causes of Unreadable Cards report
styles	*electionName*	Ballot Styles report
contests	*electionName*	Contests report
log	*electionName*	Election activity log
weblog	NA	Web activity log
about	NA	Displays a list of all files that comprise the ClearCount software, including the full file path name, the file's creation date, and a secure hash of the file contents.
stmt(1)	*electionName*	Statement of Votes Cast report
stmt(2)	*electionName*	Statement of Votes Cast by Precinct report
stmt(3)	*electionName*	Statement of Votes Cast by Precinct and Counter Group report
stmt(4)	*electionName*	Statement of Cards Cast by Precinct report
stmt(5)	*electionName*	Statement of Cards Cast by Precinct and Counter Group report
stmt(6)	*electionName*	Statement of Cards Cast by Counter Group report
stmt(7)	*electionName*	Statement of Votes Cast by Party report
stmt(8)	*electionName*	Statement of Votes Cast by Counter Group report
stmt(9)	*electionName*	Statement of Votes Cast with Districts report
stmt(10)	*electionName*	Statement of Votes Cast with Districts and Counter Groups report
stmt(11)	*electionName*	Statement of Cards Cast with Districts report
stmt(12)	*electionName*	Statement of Cards Cast with District and Counter Groups report
pdfreport	*electionName*	Election PDF reports
pdfsubmit	*electionName*	Election PDF reports
xml	*electionName*	Generate election XML results report
cvr	*electionName*	Cast Vote Record report
locator	*electionName*	Locator report yielding physical locations of ballots

9.2.13.4 URLs that produce graphic browser pages

Action	Parameters	Item
thumbs	*electionName*	Card images
ovals	*electionName*	Vote Visualization
side	*electionName*, *imageFileName*	Displays an image and information about one side of a ballot.
ballot	*electionName*, *ballotID*	Shows a visually annotated ballot.
overlay	*electionName*	Generates a test page with all ovals overlaid

9.2.13.5 URLs that are highly interactive browser pages

Action	Parameters	Item
admin/db	NA	Election Administration page
admin/db/	*commandName*	Processes a specific administration command
admin/backup/	*electionName*	Starts an election database backup
admin/startrestore	NA	Starts an election database restore
admin/startmerge/	*electionName*	Starts an election merge
admin/withdrawn/	*electionName*	Administers withdrawn contests/choices
admin/user	NA	User Administration page
admin/user/	*commandName*	Handles a specific user administration command
remaking/	*electionName*	Displays a list of ballots to be resolved
remakeballot/	*electionName*, *ballotID*	Resolves one ballot
jobs	NA	Displays background jobs

9.2.13.6 URLs that produce a naked bitmap (not an HTML file)

Action	Parameters	Item
image	*electionName*, *imageFileName*	Displays raw card image.

9.2.13.7 URLs that produce other raw files that are output of job execution

Action	Parameters	Item
jobresult	job_id	Job results files

9.2.13.8 URLs issued by browser Javascript AJAX requests

Action	Parameters	Item
get_style_info	*electionName*, *ballotStyleID*	Handles AJAX request for information on ovals for a given style

9.2.13.9 API URLs submitted by programs only (no browser, no UI)

Action	Parameters	Item
authenticate	*userName*, *password*	Authenticates a user name/password pair
writelog	*electionName*	Writes a message to the election log
writerecords	*electionName*	Writes tabulation records to the election database
readcontext	*electionName*	Returns the database BDF tables
upload	*electionName*, *imageFileName*	Uploads a card image file to the server. This URL is sent by the Tabulator application running on the ScanStation. *electionName* is the name of the election. *imageFileName* is the image data, along with the name and creation date of the file.
deletebox	*electionName*, *boxID*	Deletes a box of cards from the election. This URL is sent by BallotDeleteBox.exe running on a ScanStation or an election administration station.
querytabulated	*electionName*, *scanComputerName*	Returns the set of BallotIDs already tabulated