NIST Voting Technology Series
NIST VTS 100-5

Designing Usable Audio for Voting Systems:

Best Practices and a Test Approach

Lynn Baumeister
Whitney Quesenbery
Sharon Laskowski

This publication is available free of charge from:
https://doi.org/10.6028/NIST.VTS.100-5

NIST NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Voting Technology Series
NIST VTS 100-5

Designing Usable Audio for Voting Systems:

Best Practices and a Test Approach

Lynn Baumeister
Whitney Quesenbery
Center for Civic Design

Sharon Laskowski
Information Access Division
Information Technology Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/NIST.VTS.100-5

January 2025

U.S. Department of Commerce
Jeremy Pelter, Acting Secretary of Commerce

National Institute of Standards and Technology
Craig Burkhardt, Acting Under Secretary of Commerce for Standards and Technology and Acting NIST Director

NIST Technical Series Policies
Copyright, Fair Use, and Licensing Statements
NIST Technical Series Publication Identifier Syntax

Publication History
Approved by the NIST Editorial Review Board on 2024-05-22

How to Cite this NIST Technical Series Publication
Baumeister L, Quesenbery W, Laskowski S (2025) Designing Usable Audio for Voting Systems: Best Practices and a Test Approach. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Voting Technology Series (VTS). NIST VTS 100-5. https://doi.org/10.6028/NIST.VTS.100-5

NIST Author ORCID iDs
Sharon Laskowski: 0000-0003-2024-868X

Contact Information
voting@nist.gov

Abstract

Voting systems must support multiple interaction modes: presenting information both visually and auditorily, accepting navigation and selections from both screen touches and key presses on a tactile controller. The best practices outlined in this document focus on the audio and tactile controller experience. These best practices are intended to give voting system designers methods to improve their existing audio or for creating audio for a new voting system. These best practices describe an incremental approach to testing the audio with voters with disabilities, before the audio has been programmed into the system and is potentially more difficult to change, thereby providing an opportunity to improve the audio experience and give those voters an optimal voting experience.

Keywords

Accessible voting, audio ballot, usability testing, user-centered design, disability.

Table of Contents

Summary **1**

Introduction **2**

VVSG 2.0 requirements for accessible audio 2
Accessible voting and modes of interaction 3

Designing audio for voting systems **4**

Best practices **6**

Best practice #1: Voice text for current focus and relevant information needed for context 6
Best practice #2: Use consistent syntax 7
Best practice #3: Order information consistently 8
Best practice #4: Use pauses to break audio into meaningful chunks 9
Best Practice #5: Use earcons (nonverbal sounds) to communicate feedback 10

Designing audio for ballot screen types **11**

Designing audio for contest screens 11
Navigating the ballot 12
Reviewing and casting (or printing) the ballot 12
Voter preference screens and controls 13

A method of scripting the audio **14**

Steps for creating the script 14
How to read the audio script table 16

A method of testing the audio script **19**

The setup 19
Considerations for the computer voice 20

Wrap-up **21**

Appendix **22**

Sample script for single choice contest **23**

Summary

The best practices in this report provide guidance for designing the audio for an accessible voting system to meet the requirements of the Voluntary Voting System Guidelines (VVSG) 2.0, published by the Election Assistance Commission [1] and improve the audio experience for voters relying on voiced information.

On the web and in most applications, people with disabilities can access software using tools such as screen readers that they have installed on their computer. Screen readers both voice the information and functions on the screen and provide a keyboard-driven tactile interface to interact with the software.

Voting systems are different, more akin to a kiosk – self-contained and using a customized operating system. For security, most personal assistive technology, including conventional screen readers, cannot be used. This means that the voting system itself must provide the audio 'display' and tactile controls for interacting with the ballot.

Voting systems must also meet the VVSG 2.0, which includes requirements for the technical properties of the audio, requirements to ensure equivalent and consistent voter access and privacy, and requirements that support the mandate in the Help America Vote Act (HAVA)[2] to make independent use by voters with disabilities possible.

This report contains guidance for how to design and test accessible audio for a voting system interface as follows:

- Coreore best practices that are general design principles for how to use syntax, pauses, order, and nonverbal sounds to communicate information and actions effectively.
- Special considerations for specific functions and page/screen types in a typical voting system.
- Guidance for designing the audio that demonstrates a way to use a chart or audio script to plan the audio design to cover all interaction possibilities.
- Guidance for testing that describes a way to use the audio script along with the visual interface both for interaction walk-throughs and usability testing, even before the audio functions are programmed, making it possible to follow best practices for a user-centered design process[3].

[1] https://www.eac.gov/voting-equipment/voluntary-voting-system-guidelines
[2] https://www.govinfo.gov/content/pkg/PLAW-107publ252/pdf/PLAW-107publ252.pdf
[3] as required in section 2.2 of the VVGS 2.0

Introduction

The Voluntary Voting System Guidelines (VVSG) 2.0 requires information and instructions for a voting system to be written in plain language. This report extends that concept by describing best practices for designing audio for a voting system. It outlines principles and techniques to make an audio-tactile interface that is consistent and easy to use. A good design for the audio interface supports voters with disabilities as they mark, verify, and cast their ballot efficiently and accurately, giving them a better voting experience.

VVSG 2.0 requirements for accessible audio

The VVSG 2.0 requirements are the starting point for accessible audio best practices. Two fundamental requirements are 5.1-A – Voting methods and interaction modes and 7.2-A – Display and interaction options. Together they require that visual and audio display formats have the same functionality and that each (along with the two interaction modes of touch and tactile controls) provide full functionality to "enable voters to mark their ballot to vote, and verify and cast their ballot, supporting the full functionality in each mode."

Other requirements elaborate on specific areas in which the audio format must coordinate with the visual presentation or make all ballot and contest information available. They include:

- 5.1-A – Voting methods and interaction modes
- 5.2-A – No bias (ballot options)
- 5.2-C – Information in all modes (contest options and messages)
- 5.2-D – Audio synchronized (with visual display)
- 5.3-E – Sound cues (coordinated with visual cues)
- 7.3-K – Warnings, alerts, and instructions (and how they are separated in a list)

Finally, a group of requirements list specific types of ballot information that must be made available in both visual and audio formats:

- 7.2-C – Voter control (and what the system must announce for various selections)
- 7.3-C – Contest information (and number of choices allowed)
- 7.3-E – Feedback (confirming voter selections)
- 7.3-F – Correcting the ballot (opportunities before ballot is cast)
- 7.3-H – Overvotes (notifications if too many selections are attempted)
- 7.3-I – Undervotes (notifications if there are unused selections)

This report does not cover how to meet the technical requirements for the audio including settings, preference, and defaults in 6.1-D, 7.1-A, 7.1-B, 7.1-K, 7.1-L, 7.1-M, 7.2-G, or the hardware requirements in 8.1. Note that the audio option must also be provided for any required alternative languages as described in 5.2-B and 7.3-M.

This report focuses on interaction, providing best practice guidelines for how audio should be written, structured, and organized. It does not discuss the choice to use synthetic or natural speech as that is a technical implementation discussion.

Accessible voting and modes of interaction

The VVSG 2.0 identifies two display formats: visual and audio, and two interaction modes: using a touch screen and using tactile[4] keys (also called a tactile controller). Two common combinations are:

- Visual + Touch
  - Read information and voting options from the screen
  - Use a touch screen to navigate and make selections
- Audio + Tactile
  - Use audio to hear information and voting options
  - Use a tactile controller to navigate and make selections

Some voters, however, mix and match the formats and interaction modes depending on their personal preferences or disability.

- Visual + Tactile. Used by people with vision, but limited or no use of their hands. This mode is also helpful for voters who cannot reach the screen.
- Audio + Visual + Touch. Listening to the audio while looking at the screen, to help with language access or reading comprehension
- Audio + Visual + Touch + Tactile. Some voters may use all of the options simultaneously, using touch or gestures along with the tactile controller to navigate and make selections

The best practices in this report can help address the VVSG 2.0 requirement that the visual and audio interfaces be synchronized when both are active (for example, a voter listening to the audio while looking at the screen).

Each voting system has its own tactile controller – a collection of buttons and unique layout of the controls, but most include at least five buttons: a distinctive "enter" key for selecting/unselecting and "up", "down", "left", and "right" arrows for navigating.

Figure 1: 5 tactile buttons used in examples

ENTER UP DOWN LEFT RIGHT

The tactile controller might be physically integrated into the voting system case, but is often attached with a cable that allows it to be placed in a convenient location for each voter. Under VVSG 2.0, voting systems may allow a voter to plug in their own tactile controller equivalent such as a sip-n-puff controller to allow the voter to use custom controls.

Voters with limited use of their hands may use only two or three buttons. In this case, one button is used for selection, the other for navigation. Ideally, the audio interface should be aware of the keys available from the tactile controller and provide instructions that match the controls available.

[4] Tactile controls are designed to be found, identified, and activated using the sense of touch without requiring vision to locate or identify them. Not to be confused with touch screen controls, which require vision to use.

Designing audio for voting systems

Just as the visual design of a voting system employs positioning and appearance to group similar elements and create semantic meaning, the audio provides an equivalent structure. The audio does more than just voice text. It can use timing, sequence of information, and pauses to communicate structure and meaning.

The audio performs two important tasks:

- Voicing information presented visually on the screen.
- Communicating navigation options available from the tactile controller.

When viewing a screen, a sighted voter sees **focus**, **status**, possible **actions**, and **outcomes**. A voter listening to audio cues will need the same information.

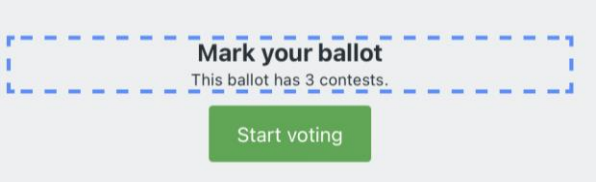
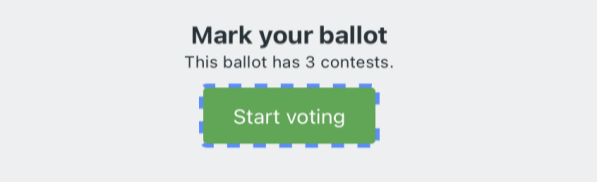
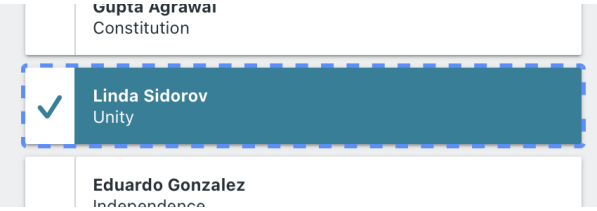
- The **focus** refers to the item that the cursor is on. In a voting system, this may be a block of text (such as the contest title or instructions), contest options (such as a candidate's name), or a button.
- The focused control may have a **status**, or a condition that depends on a voter's actions, such as whether a candidate name has been selected or is not selected.
- **Actions** define what happens when a key on the tactile controller is pressed, given the current focus and status. The difference between using touch versus the tactile controller is that a voter using touch can touch anywhere on the screen whereas a voter using a tactile controller is constrained to the keys on the tactile controller.
- The **outcome** – changes that occur as a result of an action (e.g., a candidate is selected or unselected, a contest is fully voted) and where the focus moves to (if it moves). In some instances, no changes will occur.

About the screenshots

Because the audio and visual interfaces must be synchronized, the visual design must show which screen element has the focus because instructions for using the tactile controller will be based on the current focus point.

In this report, screenshots illustrate how the screen of a voting system might look at particular moments in the voting process in order to provide context for the discussion of what the audio needs to provide. The illustrations use a dashed blue outline to show where the focus is.

Figure 2: Examples of screen elements with focus

Screen element with focus	Visual example
On a chunk of text	
On a button	
On a selected candidate	

Best practices

These best practices are basic principles for designing the audio and making decisions about what to include and how to present the information effectively in audio form. For each of the best practices we describe the guideline, how it helps voters relying on audio, how to implement it, and where relevant, provide examples.

The best practices are:

1. Voice text in the current focus and relevant information needed for context
2. Use consistent syntax
3. Order information consistently
4. Use deliberate pauses
5. Use earcons (nonverbal sounds)

Best practice #1: Voice text for current focus and relevant information needed for context

The audio must voice text contained in the screen element that has the current focus. It must also voice relevant information that is available visually and provides context.

Why it matters

Voicing the text contained in the screen element provides equivalent information to what a sighted voter can read. Voicing the relevant information provides additional contextual information a sighted voter can infer from other information on the screen.

How to implement this best practice

For each focus point:

- Voice the text being displayed in the focus
- Identify screen changes or other information that is visually perceivable and relevant to the focus.
- Incorporate that information into the audio script.

Figure 3: Introducing a contest

The text of the current focus is *"State of Hamilton Governor."*

A sighted voter can see how many candidates there are. For this focus, the audio should also voice the number of candidates, e.g., "*There are 4 candidates in this contest."*

The complete audio for this focus combines the text of the focus with the additional information about the number of candidates: *"State of Hamilton Governor. There are 6 candidates in this contest."*

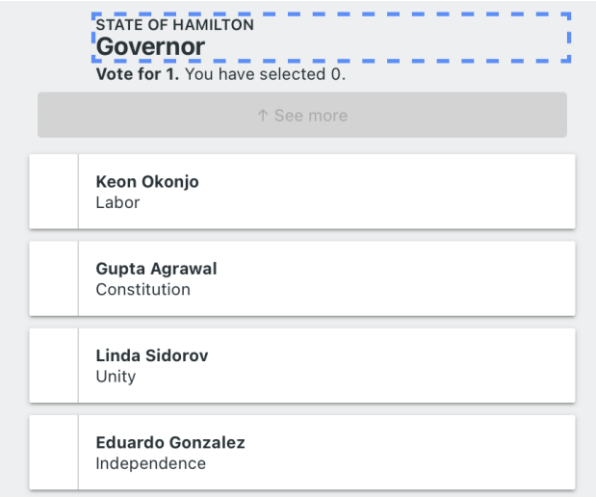
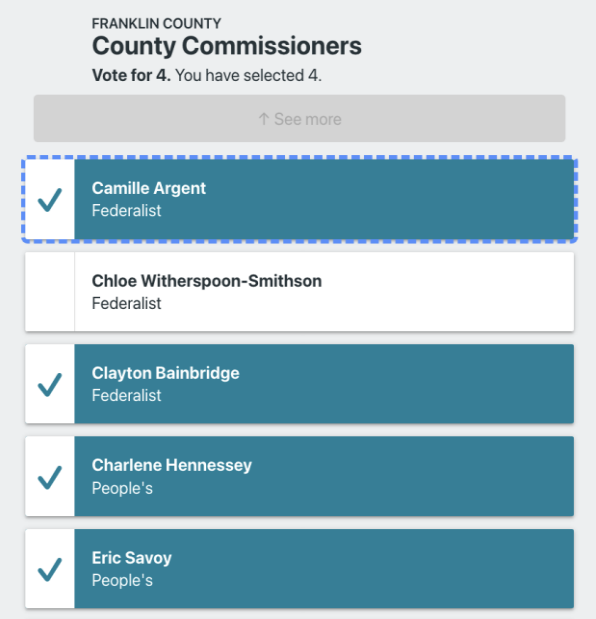


Figure 4: Candidate selection status

The voter has just selected Camille Argent. The audio needs to voice the focus and its status *"Camile Argent Federalist has been selected."*

The other relevant information available visually is that the contest is fully voted. *"This contest is now fully voted."*

The complete audio for this focus combines the text of the focus with the additional information about the number of candidates: *"Camille Argent Federalist has been selected. This contest is now fully voted."*



Best practice #2: Use consistent syntax

Use a consistent syntax for how candidates and voting instructions are voiced.

Why it matters

Predictable audio patterns minimize the effort needed to decipher meaning. A consistent syntax helps voters focus on voting, not on understanding instructions for making selections and navigating.

How to implement this best practice

- Put the outcome first, then the action to get that outcome

Voters are listening for the outcome they want. Voicing the outcome first lets voters identify the one they want before being told how to achieve it.

Figure 5: Outcome → Action Syntax

Outcome	Action to take	Audio script
Select a candidate	Press ■ Enter	*"To select candidate, press Enter"*
Continue to the next selection or function	Press ▼ Down	*"To continue, press Down"*

Best practice #3: Order information consistently

Frequently, audio will need to convey multiple pieces of information (e.g., status of the focus, possible actions, how to move on). Use a consistent order for these items:

focus → condition/status → specific actions → general actions

Why it matters

When audio is presented in a consistent order, voters need less effort to recognize and remember the information. For example, by starting with the most immediately relevant actions and then moving to the least common, voters can choose how much of the instructions to listen to. For example, if the first instruction in the collection is the action they want to accomplish they do not have to listen to the rest of the instructions.

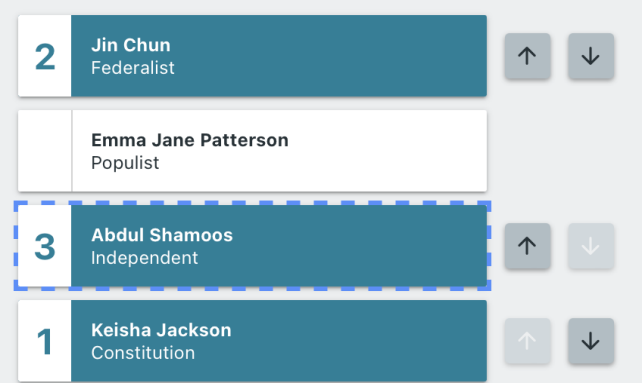
How to implement the best practice

- **Go from specific to general**
  Start with voicing the current focus, primary outcomes for that focus and status, and then continue to more general or secondary outcomes.

  focus → condition/status → primary outcomes → secondary outcomes

Figure 6: Sample script from a ranked choice voting contest

- Numbers show ranks for candidates ranked so far
- Candidate name and party listed for each option
- Color highlights each ranked candidate
- Up and Down arrows next to ranked candidates (buttons are disabled for the first and last ranked candidate)



Placeholder elements incorporated into the audio	Focus is Candidate-Party *"Abdul Shamoos Independent"* Status is Rank # *"Is ranked number 3"*
Primary and secondary outcomes and actions for each of them	Primary outcomes → actions: *"To make ranking adjustments, press RIGHT"* *"To remove ranking, press ENTER"* Secondary outcome → action: *"To continue, press DOWN"*
Audio script	*"Abdul Shamoos Independent is ranked number 3. [.] To make ranking adjustments, press RIGHT. To remove ranking, press ENTER. [...] To continue, press DOWN"*

Best practice #4: Use pauses to break audio into meaningful chunks

Break up audio with short pauses to identify transitions between different types of information.

Why it matters

Strategic pauses allow voters to chunk information and navigate an audio environment efficiently. When processing a stream of verbal or auditory information, the pauses give voters time to digest the information they've just heard before having to process the next section of audio. The pauses are analogous to separators in written text such as commas, periods, and line breaks. They provide a moment in which a voter can take action without interrupting the voicing.

Without pauses voters have a higher cognitive load digesting a continuous stream of information. As voters gain experience with the system over a single session, they learn, for example, that a long pause signals that more routine instructions are coming. Consistently placed short and long pauses reduce the total time it takes for voters to cast their ballot using an audio system.

How to implement this best practice

- Use no more than 2 types of pauses: one short and one long. The exact duration of these pauses are irrelevant, as long as they are distinguishable from each other, and neither pause is so long that the voter believes that the audio is finished (which may prompt them to move to the next screen too early).
- Use short pauses to separate information that is specific to this moment. The short pause can be 1 second.
- Use longer pauses to indicate that the contest-specific audio is complete, and the "routine" audio (additional possible actions) is about to begin. The longer pause can be 3 seconds.

Figure 7: Sample script with deliberate pauses

"Eduardo Gonzalez Independence is now selected. This contest is now fully voted. [. . .] To continue, press DOWN."

Best Practice #5: Use earcons (nonverbal sounds) to communicate feedback

"Earcons" are short pieces of nonverbal audio that can provide supplemental clues to verbal audio, give feedback on actions, or prompt voter actions. Some examples of earcons that could be used in a voting system include:

- Error noise ("beep") if the voter tries to perform an action not allowed by the system
- Success noise ("ta da") when a contest is fully voted.

Why it matters

Commonly used earcons (such as an error noise) can be processed more quickly than a spoken sentence.

How to implement this best practice

- Use no more than 2 or 3 earcons. Use sounds that are commonly used rather than inventing new sounds that may not be familiar to voters.

Figure 8: Example of using earcons

P**ressing the left key or the right key is not a valid option when** the focus is **on the contest title**. An error noise is sounded if the **voter presses either of those keys**.

Designing audio for ballot screen types

Voting systems have several kinds of screens (screen layouts) many of which are used more than once, each time showing different information. For example, all the single-choice (vote-for-1) contests will use the same layout but each will present a different contest title ("County Executive", "Governor", etc.) and candidates.

Designing audio for contest screens

The most common screens of a voting system are those that present a contest. The list below goes through contest types and key considerations when designing audio for that type of contest.

For all contests, the audio needs to voice:

- The contest title
- How many selections are allowed
- Instructions for voting the contest using the tactile controller
- How to move between candidates
- How to select and unselect a candidate
- Warnings or messages (and how to continue after voicing them)

Along with contextual information visually available on the screen such as:

- How many candidates there are
- When the contest is fully voted or progress in selections

Additional information is needed in different contest types:

- Single-choice (Vote-for-1)
  - How to change a selection
- Multiple-choice (Vote-for-N)
  - How many choices are still remaining after each selection
- Yes/No
  - The text of the question (including how to skip voicing it)

Ranked choice voting (RCV)

RCV contests are more complex. An RCV contest will present a list of options. Voters numerically rank the options in the list with the rank of #1 indicating their most preferred option, #2 indicating their next most preferred option and so on. The options are typically candidate names.

The audio needs to voice:

- The contest title
- Instructions for voting the contest using the tactile controller
- How to move between contest options
- How to rank a candidate
- How to un-rank a candidate
- How to change the rank of a previously ranked candidate
- When all the candidates have been ranked

Navigating the ballot

In addition to voting the contests, voters also need to know how to navigate other screens in the voting system.

- **Opening screens**

  The voting system may have one or more screens before the first contest. Audio must voice all text on those screens. If there are images on the screen that are purely decorative they do not need to be voiced. If the information in an image is information the voter needs to know, voice the information represented by the image.

- **Introductory, instructional, or section screens**

  Some voting systems use instructional or title screens to separate sections of the voting process and give instructions for the next phase. These are helpful for sighted voters – giving a clear signal that they are transitioning to a different phase in the voting process and especially helpful to voters relying on audio who won't see visual changes to the screen when shifting to the next phase. There are typically few actions on these screens outside of a button to continue.

- **Moving from contest to contest**

  In order to optimize voter understanding and efficiency, the design of tactile controller navigation must be consistent within the voting system. For example, there is potential for confusion if the means for moving to the next contest sometimes requires navigating to the NEXT button and sometimes requires pressing the RIGHT key. The design of the tactile controller navigation must be consistent within contests and consistent when moving between other elements (between contests, getting to buttons in the frame such as Help, Settings, Print, etc.)

- **When buttons change label (Skip, Return to review, etc.)**

  Sometimes a screen will have more than one version depending on how the voter reached the screen. For example, in some voting systems, the button at the end of the contest is labeled "Next" (because it moves to the next contest) but if the voter reaches the contest screen from a review screen, the button at the end of the contest is "Return to review" instead of "Next" because the button will move the voter back to the Review screen.

  The audio must state those changes, accurately reflecting the text on the screen at all times.

- **Dialogs / overlays**

  A non-sighted voter will not have the same visual cue that a sighted voter has that a dialog overlay has popped up on the screen. The audio needs to cue the voter that there is a dialog box because the presence of a dialog box provides context for the next set of audio. Within a dialog box or overlay the same best practices apply: voice the focus, announce status changes, and tell the voter what outcomes are available and how to achieve those (what key press to use).

Reviewing and casting (or printing) the ballot

After the last contest a voting system will show a review screen. The audio will need to provide equivalents for information and options available from the review screen. Those options will include:

- Moving through the review screen to hear each contest and how the voter marked it – including if the voter chose not to vote the contest.
- Communicating how to change their vote.

  Typically, this action will move the voter back to the screen for that contest. If any of the buttons have changed due to the voter coming into the contest from the Review screen (for example the Next button has been renamed to "Return to Review Screen"), the audio must reflect that change.

The final phase of a voting system will either be casting the ballot directly from the voting system or printing the ballot for further processing (e.g., to be fed into a scanner at a polling place, or mailed to an election office). This process will typically have dialog boxes or screens confirming the voter's intention to proceed with casting or printing their ballot.

Final steps may include physical actions such as ability to review a printed ballot before the machine casts it, or picking up a printed ballot from an adjacent printer. The audio may need extra text, beyond what is on the screen, to fully describe the final steps to a non-sighted voter. A non-sighted voter will need to be told the location of objects and controls, such as the location of the printed ballot or physical controls to complete the voting session.

Voter preference screens and controls

All voting systems include options for voter preferences. The VVSG 2.0 (7.1-K – Audio settings) requires options for the speech rate and volume. Some voting systems also include settings such as pitch, or gender. These settings may include controls that are different from the contest screens, such as sliders, or multiple selection of settings preferences.

When designing instructions for using the tactile controller:

- Break the instructions into short chunks.
- Give the voter control over when to move from chunk to chunk.
- Describe the location for any plugs or controls on the voting system itself.
- Describe buttons by shape, location, and color.
- Describe preference functions that are on the tactile controller such as buttons for adjusting the volume.

Examples:

"*The enter key is a red rectangular key in the middle of the keyboard. To the right of the enter key are four orange triangle-shaped directional keys, left, right, up, down [.] to hear the next instruction, press down key*".

"*To change the volume at any time, use green diamond-shaped keys in the lower right corner [.] to hear the next instruction, press the down key*".

A method of scripting the audio

The audio script is a way of identifying all possible combinations of focus, status, and actions (key presses) from the tactile controller so that the audio knows how to respond to any possible interaction. A methodical way to capture the audio requirements is a step-by-step process that starts at a high-level, identifying the different kinds of screens and then identifying focus points, different status the focus can have, and for each of the keys on the tactile controller, what happens (the outcome).

If the voting system can detect when an alternate controller is plugged in, such as a voter-supplied 2-switch controller (such as a sip-and-puff mouse) or a 3-switch controller, then the voting system can switch to an alternate audio script optimized for that type of controller.

The steps below can be applied to any voting system. The screenshots and accompanying audio script are for purposes of illustration. The examples of tactile controller actions (buttons that can be pressed) is limited to 5 tactile keys (ENTER, LEFT, RIGHT, UP, DOWN).

The goal when designing audio is to give the voter the information they need to vote according to their intentions without overwhelming them by voicing all the tactile controller options at each focus point. For example, if pressing the LEFT or RIGHT key from a certain focus point will result in an error earcon (because that action doesn't do anything from that focus point), don't include those keys in the audio for that focus point.

Steps for creating the script

- **Identify the kinds of screens** that are present in the voting system (e.g., start screen, interstitial screen, single choice contest, vote-for-N contest, yes/no contest, Help, Settings, Review).

  For example, if a voting system has 3 interstitial screens (screens that separate sections of the voting process) but they all have the same layout (explanatory text + a single button) that is one kind screen with different text on each version of the screen. Likewise, all of the vote-for-1 contests will use the same layout, just with different contest titles and candidates.

- For each kind of screen, **list all places on the screen that can have the focus**.
  - Buttons
  - Text such as
    - Contest title (e.g, "State of Hamilton, Governor", "Question A:")
    - Instructions ("Vote for 1; you have voted for 0", "Vote yes or no")
    - Question text ("To upgrade educational facilities at Diablo Valley, and Franklin, Colleges, and the San Brentwood center, and help prepare students for jobs and college transfer by modernizing classrooms and labs, building facilities for health, medical, science, and technology training, and implementing infrastructure improvements, shall the Franklin Community College District issue $450 million of bonds at legal interest rates with independent oversight, audits, and all funds spent on local sites?")

  - Voting options such as
    - A candidate ("Jamshid Sahila")
    - Yes

    - No
    - A write-in box

- **For each focus point on a screen**:
    - Make a list of the possible conditions the focus can have (e.g., candidate selected, candidate unselected).

      Include screen changes that happen in other parts of the screen that will need to be voiced (such as updating how many in a vote-for-N contest have been voted)

- **For each focus/status combination**:
    - List the outcome for each possible action from the tactile controller (e.g., pressing the LEFT key, pressing the RIGHT key)

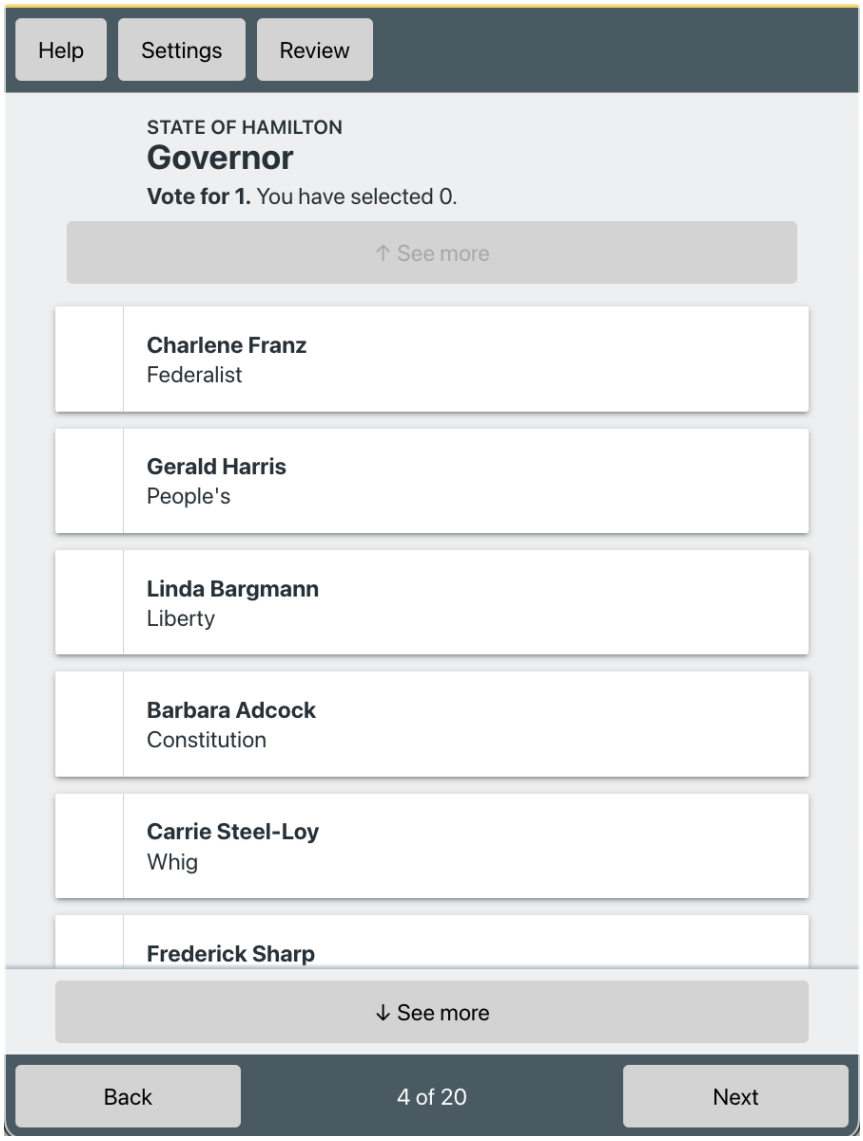
      The reason for listing all possible actions, not just the obvious ones such as going to the next candidate, is to ensure that none get missed, so that the system will know what to voice no matter what action the voter takes.

      For each outcome capture:

        - Changes to the focus (where the focus moves to). There may be situations where the focus doesn't change as the result of a tactile controller press.
        - Changes to the status of the focus (such as when a candidate changes from unselected to selected or vice versa). Some foci won't have a status.
        - Conditions that affect the outcome (such as the focus is on the first candidate, or the last candidate of a contest)
- Once this information has been compiled, **use the information gathered from those steps and the best practices to generate the audio**, including static text and placeholders for each focus/status combination. Placeholders indicate where the system will fill in the particulars of the contest (e.g., candidate names, text of referendum question, contest title)

The next section walks through an example of an audio script for the screenshot in Figure 9: Single-choice contest.

Figure 9: Single-choice contest

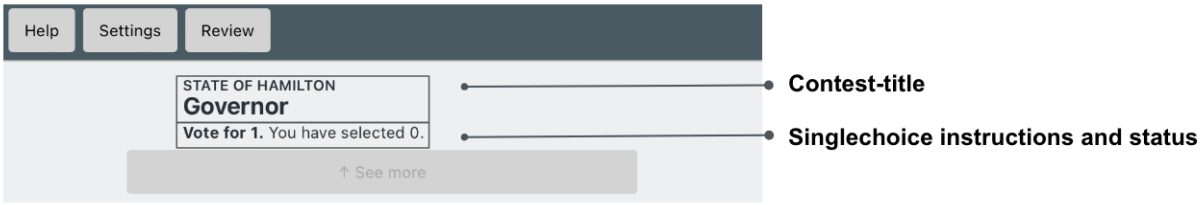


How to read the audio script table

This list describes the columns in Figure 11: Audio for a single choice contest (first 3 rows).

- The **Focus** column contains the name given to a location on the screen that can have the focus. Naming the focus points makes it possible to have the audio script refer to it easily. It is helpful to use a name that is indicative of what that focus point is. Figure 10: Sample focus point names shows what we named two of the focus points on a single choice contest screen (Contest-title, Singlechoice instructions and status).

Figure 10: Sample focus point names



- The **Status** column indicates the different status the focus might have (e.g., selected, not selected). In some rows this column is blank because the focus point doesn't have a status.
- The **Audio** column is what the audio will voice. Sometimes what is voiced is static text (always the same) such as instructions for a yes/no contest "Vote yes or no", but many times the text is variable depending on what contest is being shown. In the audio column a focus surrounded by ( ) indicates that it is a placeholder. Rather than voicing the focus name, what should be voiced is the actual text associated with the contest.

  Example from row 1 in Figure 11: Audio for a single choice contest (first 3 rows)

  - What the audio column contains:

    (Contest-title). There are **(#)** candidates. To navigate through candidates, use the up and down keys. To select or unselect a candidate as your vote, use the enter key. [. . .] To continue, press down.

  - What the audio will voice if the contest is the governors contest shown in Figure 9: Single-choice contest.

    "State of Hamilton Governor. There are 12 candidates. To navigate through candidates, use the up and down keys. To select or unselect a candidate as your vote, use the enter key. [. . .] To continue, press down."

- The **Possible actions and outcomes** column lists all the possible key presses. The tactile controller used in this script sample has 5 keys:

  ◀ Left

  ▶ Right

  ▲ Up

  ▼ Down

  ■ Enter

Figure 11: Audio for a single choice contest (first 3 rows)

This page has just the top three rows of sample script for a single choice contest such as the one in Figure 9: Single-choice contest. To view the full sample script see Appendix: Sample script for single choice context.

Focus name	Status	Audio	Possible actions and outcomes	
Contest-title Text block		***(Contest-title).*** *There are* ***(#)*** *candidates. To navigate through candidates, use the up and down keys. To select or unselect a candidate as your vote, use the enter key. [. . .] To continue, press down.*	◀	no focus change, error noise
			▶	no focus change, error noise
			▲	focus moves to Review button
			▼	focus moves to **Singlechoice instructions and status**
			■	no focus change, repeat audio
Singlechoice instructions and status Text block		*Vote for one. You have selected* ***(#).*** *[. . .] To continue, press down.*	◀	no focus change, error noise
			▶	no focus change, error noise
			▲	focus moves to **Contest-title**
			▼	focus to moves to first candidate
			■	no focus change, repeat audio
Candidate-party Text block	Not selected	***(Candidate-party)*** *is not selected.[. . .] To continue, press down.*	◀	no focus change, error noise
			▶	no focus change, error noise
			▲	if focus is on first candidate, focus moves to **Singlechoice instructions and status** Otherwise focus moves to previous candidate
			▼	if focus on last candidate, focus moves to **Next** button Otherwise focus moves to next candidate
			■	If no other candidate is selected, focus changes to **Candidate-party + Selected** If another candidate is already selected, focus moves to **Overvote**

A method of testing the audio script

One of the biggest challenges for designing audio is being able to test it early enough to make changes in response to what you learn in the testing. This section outlines a method for testing the audio before it is programmed into the voting system. Testing early, before it is programmed in, allows for confirming that the script supports voters (they are able to navigate and make voting selections without confusion or uncertainty) and experimentation (observing what effects the changes have on voters relying on the audio). At this testing stage, the audio should be in a form that is easily changeable.

Testing is usually done with a working preliminary system or an interactive prototype. This is an ideal way to test because you can observe how the voter responds to the system in real time. This method allows for preliminary testing with a semi-interactive prototype by having a live person on the testing team act as the voice of the system, reading from the script [5]. This allows usability testing with actual voters before the system is completely developed and it is still easy to make changes.

This testing is part of meeting the user-centered design process outlined in Principle 2 and usability testing required in Principle 8 of the VVSG 2.0.

The setup

A three person test team is needed to run the test. The test team roles are:

- **Moderator** – interacts with the participant.
- **Computer voice** – voices the script (voices the audio), based on the current focus, status of the focus, and conditions; as the participant moves through the voting system.
- **Computer/voting system operator** – runs the prototype based on the key the participant says they would press. Having a testing member run the prototype has several benefits; it allows for testing a prototype that does not yet have all the accessibility functionality operational, it provides a momentary delay which allows the computer voice to shift to the next place in the script, and it lets the moderator and voice know where they are on the screen.

Testing can be done in-person or remotely. Remote testing may make it easier to find participants who would typically use the accessibility features of the voting system because travel is not required. During a remote session, the test team shares a prototype using web conferencing software.

- The test team needs to be able to see the screen so they can keep track of the interaction, including where the focus is, what the status of the focus is and so on, because that affects what gets voiced.
- Completely blind participants don't need to see the screen and can say what control they want to use instead of using a fully-working tactile controller.
- Voters with low vision or others who use the audio, can look at the prototype as it is controlled by the computer/voting system operator, and also say what control they want to use.

[5] There are several popular methods for testing without a digital prototype or working version, generically called "paper prototyping" but the authors first heard the idea of an audio equivalent from the Los Angeles Voting Systems for All People project and a test of the audio conducted by IDEO.

This method of testing allows for iterative testing and revisions: run sessions with a few voters, review the findings, adjust the script, run sessions with a few more voters to see if the changes are beneficial, neutral, or problematic.

Testing with voters with a range of disabilities will help ensure that the audio is not tailored to one interaction mode at the expense of other interaction modes (e.g., Audio + Tactile) and is usable across a range of disabilities and interaction modes (e.g., Audio + Visual + Tactile, Visual + Audio + Tactile + Touch).

After the moderator's introduction to the session:

- The computer voice read the audio for first focus point of the prototype
- The participant said what key they wanted to press (e.g. "down")
- The computer operator interacted with the prototype and caused that keypress to happen.
- The computer voice read the audio according to where the focus had moved (based on the keypress) and what the status of the focus was now
- The participant said what key they wanted to press (e.g. "down")
- The computer operator interacted with the prototype and caused that keypress to happen.
- Etc.

Considerations for the computer voice

- The computer voice role (voicer) requires someone with the ability to read aloud with a consistent cadence and insert precise pauses as indicated by the script (e.g., voicing [.] and [...] consistently).
- The computer voice role needs the script in a form that allows them to quickly jump around each time the participant performs an action (key press). A dry run before the first session with a voter participant will help the voicer test out their ability to jump around in the script as the participant presses keys and determine whether having a printout of the script or working off of an onscreen document is most effective.
- Participants can interrupt – indicate a key press before the voicer has read the entire script for the focus. This is not an indication of rudeness – voters do this to be efficient. When voters have heard the option they want, or already know what they want to do next and how to achieve it, they interrupt and select the next keypress. When this happens, the voicer needs to stop reading the audio, move to where the participant's key press has taken them, and begin voicing that new audio.
- It is helpful for the voicer to be able to see the screen so they get a visual cue as to where the participant (and the voicer) are.

Wrap-up

Usability testing is important in ensuring that the experience using the audio presentation and tactile controller to navigate and make voting decisions is easy for voters who rely on this mode of voting. This is especially true when most of the design team is visually oriented and more familiar with the visual interface and touchscreen navigation. The best practices outlined in this document are intended to give voting system designers tools for reviewing existing audio in order to improve it or creating audio for a new voting system. Additionally, testing the audio with voters with disabilities before it has been programmed into the system and is potentially more difficult to change, provides an opportunity to optimize the audio + tactile experience, giving those voters an optimal voting experience.

Appendix

The research behind this report

The best practices in this report start from research [6] done by the Center for Civic Design (CCD) in 2020 into designing an accessible voting system that could handle a ballot with mix of RCV and non-RCV contests. The goal of that research was a design that worked for Visual + Touch, Audio + Tactile, and any combination of those interaction modes. An important aspect of the design was iterative testing with voters with disabilities to see where the design helped them, and where the design needed improvements. Part of the research was experimenting with syntax, order, and pacing to learn what was most efficacious for the participants by using a script in an easily editable form and using a person with well-modulated speech cadence to voice the audio rather than having it programmed into the system.

The best practices emerged as overarching principles. Although the screenshots and examples are specific to the voting system / tactile controller that was tested, the best practices apply to any voting system. The screenshots in this report are taken from the functional prototype used in the research. The contests show fictional candidate names followed by fictional parties (Liberty, People's, etc.). The prototype was built on code developed as a demonstration interface for ElectionGuard.

Additional Reading

The Los Angeles County project to create a new voting system, the Voting System for All People (VSAP) included extensive research with voters with disabilities and a novel approach to testing the audio experience for blind voters.
https://vsap.lavote.gov/research/

Center for Civic Design and the Ranked Choice Voting (RCV) Resource Center included audio design in a project to design an accessible RCV ballot.
https://civicdesign.org/topics/rcv/

One source of inspiration for the approach to testing audio ballots came from a method called *paper prototyping.* The book *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces* by Carolyn Snyder is a classic. It includes an excellent chapter on how to work with participants during the testing. https://books.google.com/books?id=YbzBWfTHorQC

The company User Interface Engineering wrote several articles about their work with this method:

Paper prototypes:
https://articles.uie.com/paper_prototyping/

Looking Back on 16 Years of Paper Prototyping by Jared Spool:
https://articles.uie.com/looking_back_on_paper_prototyping/

[6] https://civicdesign.org/wp-content/uploads/2017/07/Accessible-RCV-ballot-final-report-2020-0603.pdf

Sample script for single choice contest

This script is in reference to a single-choice contest as shown in Figure 9: Single-choice contest. Refer to How to read the audio script table for a description of the columns.

Main contest section of the screen

Focus name	Status	Audio	Possible actions and outcomes	
Contest-title Text block		***(Contest-title).*** *There are* ***(#)*** *candidates. To navigate through candidates, use the up and down keys. To select or unselect a candidate as your vote, use the enter key. [. . .] To continue, press down.*	◀	no focus change, error noise
			▶	no focus change, error noise
			▲	focus moves to Review button
			▼	focus moves to **Singlechoice instructions and status**
			■	no focus change, repeat audio
Singlechoice instructions and status Text block		*Vote for one. You have selected* ***(#)****. [. . .] To continue, press down.*	◀	no focus change, error noise
			▶	no focus change, error noise
			▲	focus moves to **Contest-title**
			▼	focus to moves to first candidate
			■	no focus change, repeat audio

Focus name	Status	Audio	Possible actions and outcomes	
Candidate-party Text block	Not selected	***(Candidate-party)*** *is not selected.[. . .] To continue, press down.*	◄	no focus change, error noise
			►	no focus change, error noise
			▲	if focus is on first candidate, focus moves to **Singlechoice instructions and status** Otherwise focus moves to previous candidate
			▼	if focus on last candidate, focus moves to **Next** button Otherwise focus moves to next candidate
			■	If no other candidate is selected, focus changes to **Candidate-party + Selected** If another candidate is already selected, focus moves to **Overvote**
Candidate-party Text block	Selected	***(Candidate-party)*** *is selected. This contest is now fully voted.[. . .] To continue, press down.*	◄	no focus change, error noise
			►	no focus change, error noise
			▲	If first candidate, focus moves to focus to **Singlechoice instructions and status**. Otherwise focus moves to first candidate
			▼	if focus on last candidate, focus moves to **Next** button Otherwise focus moves to next candidate
			■	focus changes to **Candidate-party + Unselected**
Overvote Dialog		*You may only select 1 candidate in this contest. To vote for this candidate you must first unselect a previously selected candidate. [...] To continue, press enter*	◄	no focus change, error noise
			►	no focus change, error noise
			▲	no focus change, error noise
			▼	no focus change, error noise
			■	focus moves to **Candidate-party + Not Selected**

Framework area of the screen

Focus name	Status	Audio	Possible actions and outcomes	
Next Button		*Next button. To go to the next contest, press enter. To go back, or for Settings, or Help, press down.*	◄	no focus change, error noise
			►	no focus change, error noise
			▲	focus moves to the last candidate
			▼	focus moves to the **Back** button
			■	focus moves to next screen
Back Button		To go back, press enter. For more options, press down.	◄	no focus change, error noise
			►	no focus change, error noise
			▲	no focus change, error noise
			▼	focus moves to **Help** button
			■	focus moves to previous screen
Help Button		To hear Help for this voting system, press enter. For more options, press down.	◄	no focus change, error noise
			►	no focus change, error noise
			▲	focus moves to **Back** button
			▼	focus moves to **Settings** button
			■	focus moves to **Help** screen

Focus name	Status	Audio	Possible actions and outcomes	
Settings Button		To hear ways to customize this voting system's settings, press enter. For more options, press down..	◀	no focus change, error noise
			▶	no focus change, error noise
			▲	no focus change, error noise
			▼	focus moves to **Review** button
			■	focus moves to previous screen
Review Button		To hear a review of your selections, press enter. To hear the contest, press down	◀	no focus change, error noise
			▶	no focus change, error noise
			▲	no focus change, error noise
			▼	focus moves to **Contest-title**
			■	focus moves to Review screen