



Board of Elections

2025 EAC Clearinghouse Award Submission: *At-Home Provisional Exercise*

Introduction

Wake County Board of Elections
Category- Poll Workers: Best Practices in Recruiting, Retaining, and Training Poll Workers

Written by the staff at the Wake County Board of Elections 1200 N. New Hope Rd. Raleigh, NC 27610

Wake County, North Carolina is home to more than a million residents, 841,883 of which are registered voters. Growing more than twice as fast as the rest of the state and adding approximately 51 people per day and over 250,000 residents in the last decade alone, Wake County has and continues to experience unprecedented growth rates. As we began planning and developing training materials for the 2025 Municipal Election, we looked back at the 2024 General, where a record-setting 600,000 ballots were cast between the 17-day Early Voting period and Election Day; of those, we received over 6,000 voters who casted their vote provisionally. This volume precedent -- along with other challenges -- has forced our team to use new tools, think outside the box, and grow our base of over 3,000 Precinct Officials to continue running efficient and secure elections amidst the emergence of new and exciting challenges.

With increased population growth comes new need: more votes means more provisional ballots being cast; more turnout requires an increase in the quantity of Officials that need to be trained; more Officials means having to start training earlier and conducting more classes (which means some Officials take their training classes two months before Election Day); having more Officials means that a dedication to providing training that works for a wide variety of learning styles is more essential than ever before. And yet, as these variables exist in constant flux, the statutory dates by which certain processes must be completed often stays the same, making it increasingly vital to iron out errors and inefficiencies to meet electoral deadlines. In addition to all of this, North Carolina legislature changed the provisional processing deadline from 10 days after the election to 3 days after the election. With these challenges, we needed to evaluate changes we could make to handle the tightened deadline and increased volume.

As a team, we've always strived to reflect and think critically about the training we offer, looking to make it more engaging and interactive in whatever ways we can. One of the ways the team chose to improve our training this year was to create the *At-Home Provisional Exercise*, a self-paced practice module that aimed to tackle two problems with one solution: first and foremost was to alleviate the unease a lot of Officials felt going into Election Day; second was the procedure retention issues that Officials have reported experiencing due to the gap between training and Election Day. Ideally, if our Officials could be better prepared to process provisionals on Election Day, our team would be able to process them faster during post-election processing due to having to spend less time on auditing and correcting mistakes.

Although the number of provisional ballots usually only accounts for around 1% of the total number cast, we fully believe in providing consistency throughout the voting experience, valuing the voice and voting rights of every single individual that walks into the polling place. As such, it is vital that every single provisional ballot be given the care and consideration it deserves. When processing provisional ballots in the days between Election Day and county Canvass, our team records every sealed provisional envelope and then conducts research using the information provided on the envelope to determine if the ballot inside may be counted or not. Due to the fact that these envelopes are not opened unless we have determined the ballot inside can be counted, the legibility, completeness, and accuracy on the envelope are crucial factors when making a determination. The *At-Home Provisional Exercise* was created to help ensure fair, equitable elections are being held, a belief that stands at the very heart of what we do at the Wake County Board of Elections.

The *At-Home Provisional Exercise* has two main components: first is an online module that can be launched through any internet web browser and contains a simulation of Voter and Street Tool (VAST), a database search program that our Help Table-trained Officials use on Election Day; second is a PDF that contains six different voter scenarios, replications of a provisional envelope, and everything else the Official would regularly use on Election Day to process these voters. The latter portion could be printed out at home and used in conjunction with the former to effectively practice the process.

The Two Components

VAST

As stated previously, VAST stands for “Voter and Street Tool”, a database tool our Information Technology (IT) team created in Microsoft Access that allows our Precinct Officials to look up registered voters and residential addresses on Election Day using distributed laptops. It contains two main sections: Voter Search (orange) and Street Search (green). Officials use this program on Election Day to confirm voters are registered, check their assigned precinct, and verify the ballot style they should receive. This software is most often used to process voters who were not in their precinct’s pollbooks, many of which choose to vote provisionally.

Type the voter's name into the fields below:

Last Name: OTTLEN (Required) First Name: ADAM (Required)

View Voters Clear

Click on voter's name to view more information:

Name	Address	Race	Age
OTTLEN, ADAM JOE	2613 VAN DYKE AVE, RALEIGH	W	33

1 result found

Go to Street Search Main Menu

VOTER

Type the voter's address as of 2/1/2026 into the fields below:

House #: 100 (Required) Street Name: MAIN (Required) City: (Optional)

View Street Info Clear

Range	Dir	Street	Type	Suffix	City	Zip
1-322	S	MAIN	ST		WENDELL	27591
Precinct: 10-03 Muni: WEN NC Sens: 14 Ward: NC House: 039 Polling Place: CARVER ELEMENTARY SCHOOL PP Address: 291 LILES DEAN RD - WENDELL, NC 27591 Cong: 02 County Com: 1 Sup Court: 10C School: 1 Ballot Style (17): D0021/R0019/L0010 Judicial: 10B Prosecutorial: 10 Ballot Style (18+): D0020/R0018/L0009						
9-317		MAIN	ST		KNIGHTDALE	27545
Precinct: 17-02 Muni: KNI NC Sens: 14 Ward: NC House: 039 Polling Place: KNIGHTDALE ELEMENTARY SCHOOL PP Address: 109 BRIDGE ST - KNIGHTDALE, NC 27545 Cong: 02 County Com: 1 Sup Court: 10E School: 4 Ballot Style (17): D0021/R0019/L0010 Judicial: 10B Prosecutorial: 10 Ballot Style (18+): D0022/R0016/L0008						
100-228	N	MAIN	ST		HOLLY SPRINGS	27540

5 results found

Go to Voter Search Main Menu

STREET

After typing into the fields on either screen, you can press the “View...” button or press the “Enter” key to search. VAST also contains an automatic wildcard function, allowing the user to get more results by typing less information. This means that typing “Melissa Car” into Voter Search will get you various results, including “Melissa Car”, “Melissa Carson”, “Melissa Carleton”, etc. It also means that leaving the city field blank in Street Search will give you multiple results if the same address exists in multiple cities within the county.

Clicking a voter’s name in Voter Search will also bring up a purple profile page that lists important demographic information for the voter, such as precinct, last-known address, Voter Registration Number (VRN), age, and Registration Date. Some of this information is used when filling out the provisional envelope while other details can be helpful in ensuring the profile is for the correct person or checking the voter’s Photo ID meets the state law requirements if they decide to vote provisionally.

OTTLEN, ADAM JOE
VRN: 0001 0051 1868

Registered at the following address:

Address: 2613 VAN DYKE AVE, RALEIGH
Precinct: 01-01
Polling Place: BROOKS AVENUE CHURCH OF CHRIST
700 BROOKS AVE RALEIGH 27607

Party: DEM
Race: W
Age: 33
Reg. Date: 9/16/2016

Back to Voter Search Go to Street Search Main Menu

Provisionals

Using the information from the green Street Search and purple Voter Profile screens, the Official records the necessary information into section B of the provisional envelope. This information is used to fill out Section C on the back of the envelope and is used during the research portion of post-election processing when determining whether the enclosed ballot can be counted.

B. Official must use laptop to complete this section.

Use <u>Voter</u> Search Screen. → Type the voter's name.	Use <u>Street</u> Search Screen. → Type the voter's residential address from Section A above.
Is the voter registered to vote in Wake County? <input type="checkbox"/> No <input type="checkbox"/> Yes - Write Voter's VRN: <input type="text"/>	<input type="checkbox"/> If voter checked NO on the 30 day question in Section A, mark this box and call the Help Line.
<input type="text"/> Voter's Registered Party Affiliation	<input type="checkbox"/> Precinct of Voter's Address in Section A OR <input type="checkbox"/> Check Box if Not in County
	<input type="text"/> Ballot Style of Voter's Address in Section A (PRIMARIES ONLY: If provisional reason is INCORRECT PARTY, write ballot style of registered party.)
	<input type="text"/> Ballot Style Issued to Voter

The Problem

Provisional Section C

In addition to the aforementioned steps, the Official must also complete the listed steps from Section C of the Provisional Envelope using the information from Section B and additional materials. Getting these items into the hands of Officials was easy enough: we simply created a packet using Adobe Photoshop and Adobe Indesign that could be printed off by Officials at home. The same could be done with an exercise answer sheet. The challenge we were faced with was getting VAST into their hands so that they could accurately practice looking up information and filling in the envelope.

VAST Limitations

As previously described, VAST was created in Microsoft Access, a program that requires a subscription to the Microsoft 360 suite or an individual purchase. Additionally, creating a training version of VAST with as many scenarios as we wanted would have been a consistent burden for our IT team to update every year as the specifications of each election type would require ongoing adjustments. These two factors simply made a distribution of a training version of VAST to our precinct Officials for individual use infeasible. We needed something that could be easily maintained, functionally equivalent to the program, and completely free for our Precinct Officials to use.

Articulating a Solution

Storyline

Storyline is a software created by Articulate that is often used for building e-learning modules. Our team has used this software for an online version of our Basic Training class since 2016. These e-learning modules are similar in structure to PowerPoint but are far more customizable and flexible, boasting the following benefits:

- Courses can be built with or without linearity, expressing content both vertically and horizontally along multiple tracks.
- Modules can be uploaded to a subscribed LMS or automatically transcribed with HTML onto the world wide web.
- Has “Triggers”: logic-based functions that do specific things when specified criteria are met. Triggers are usually built to replicate and function like traditional conditional statements, containing both a hypothesis and a conclusion: “If I go to the correct precinct, then I will be able to put my ballot inside the Tabulator.”

↑
Hypothesis

↑
Conclusion

C. Official completes this section.

After the voter has completed Section A and you have completed Section B:

1. On the **Provisional Pollbook** (pink sheet), write the voter's name, address, ballot style issued, and attach **PIN label #1**.
2. At the top of this **envelope**, attach **PIN label #2**.
3. On the **Provisional Voter Instructions**, mark the reason(s) the voter is voting provisionally, and attach **PIN label #3**. Issue the form to the voter.
4. At the top of this **envelope**, write the **precinct where voter is voting** (or Early Voting site code).
5. At the top of this **envelope**, write **voter's Provisional Pollbook number** (e.g. 01, 02, 03).
6. Get the voter's ballot from the Ballot Table.
7. In the box at the top of the ballot, write the letter “**P**” and the **voter's precinct from Section B**.
8. Fold the ballot, and place it inside the unsealed envelope.
9. Hand the unsealed envelope to the voter, and direct the voter to the provisional voting booth. Instruct the voter to return the voted ballot sealed inside the envelope to you.

After the voter has returned the sealed envelope to you:

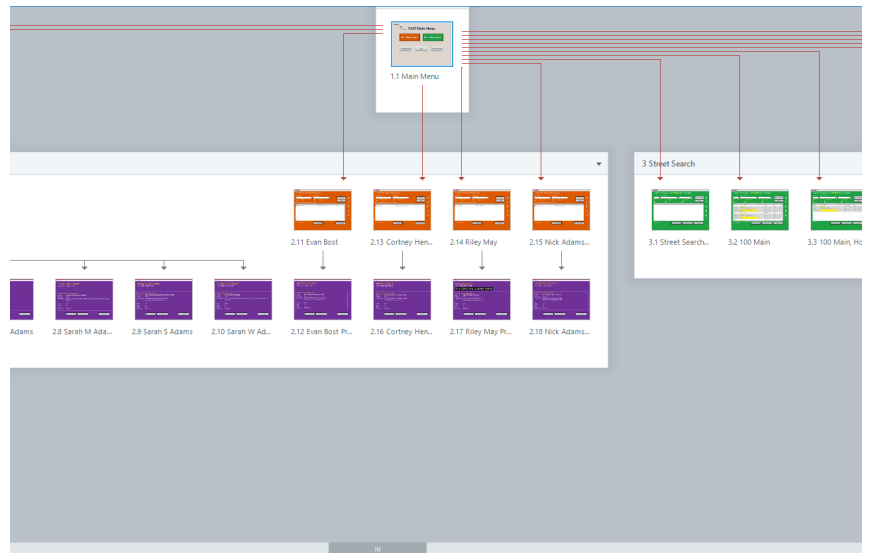
10. Review this envelope to **verify all yellow fields in Section A are complete**, and issue an “I Voted” sticker to the voter.

After the voter has left the Help Table:

11. In the boxes below, write the reason why the person is voting provisionally, print your name, and initial.

Official's Name <i>Please print legibly.</i>		Initials
--	--	----------

12. Write “VOID” on the voter's ATV or related forms, and paper clip them to this envelope.
13. Review the envelope to **verify all blue fields in Sections B and C are complete**, and place the envelope in the red plastic bag.



Using Triggers

Triggers can range widely in complexity and contain a vast array of functions and conditionals. They may:

- allow the user to jump from one slide to another when a certain condition is met
- hide or activate specific elements on a slide,
- change the state of objects,
- trigger animations,
- store information inside variables,
- and many other things.

Within a trigger, the traditional “conclusion” is represented as the “Action” while the “hypothesis” is represented by the “When” field, with the “Conditions” field sometimes being used as well.

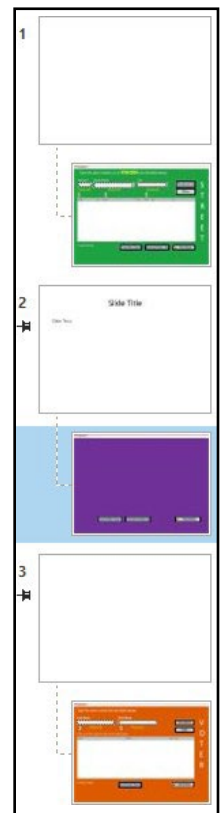
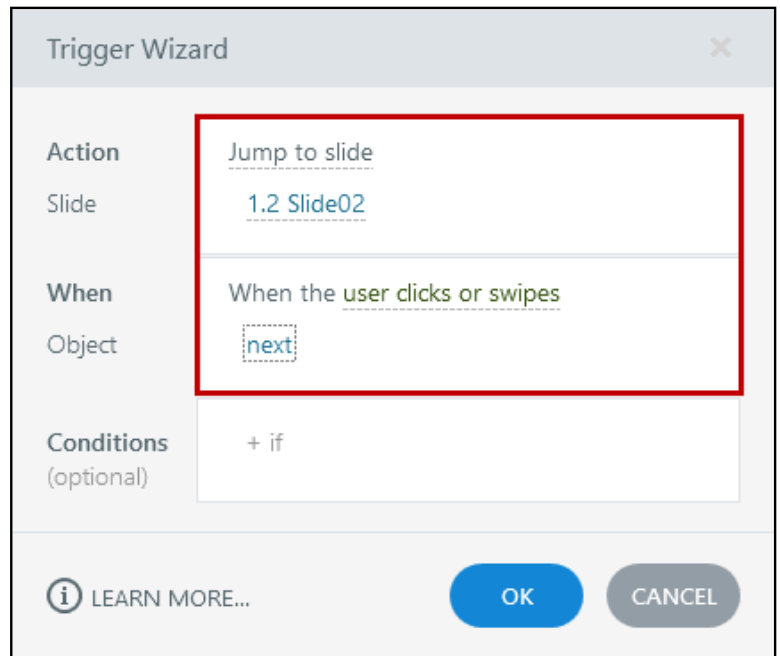
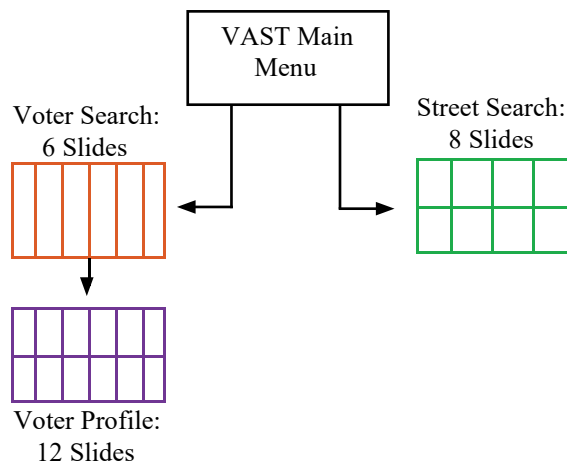
These triggers allow the user to get creative and do things that usually wouldn't be possible in other programs.

In addition, triggers can be used to activate Javascript: an object-oriented computer programming language commonly implemented with HTML to create interactive effects within web browsers. Combine this with the ability to upload straight onto the web and Storyline gave us the option to create an entirely free logic-driven online program. This opened the door for an idea: what if we created a functional 1:1 simulation of VAST with all the same functionality using triggers, Javascript, and images from the program? That intersection of features and ideas made the *At-Home Provisional Exercise* a real possibility!

The Structure

Story View

Structurally, Storyline establishes the directional and visual framework that allows the entire module to function. The VAST main menu was captured as an image and then used to create a landing page within the module. From there, clicking “Go to Voter Search” or “Go to Street Search” will bring the user to one of twenty-six possible slides, depending on stored values from the Javascript that will be discussed later. Each of those slides was built from one of the three established Master Slide layouts: voter search (orange), street search (green), and voter profile (purple). This was done so that shared functionality (e.g., “Return to Main Menu” button) and visual elements can exist without bloat and repetition. At its simplest, the module is structured with diverting paths. However, buttons also exist that allow the user to jump between the Voter Search, Street Search, and Voter Profile slides and allow the user to retain searches within each track until they click the “Clear” button. This makes going back and forth to cross-check information easy.

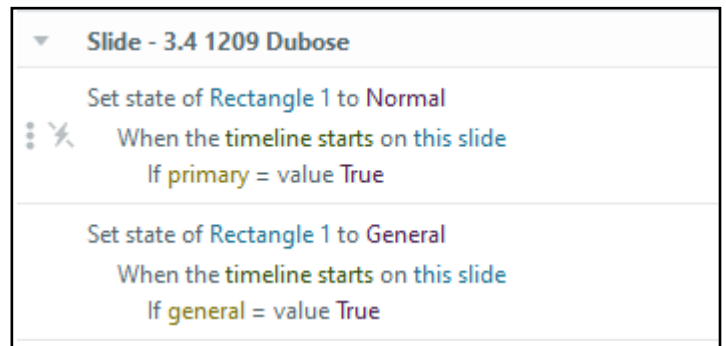
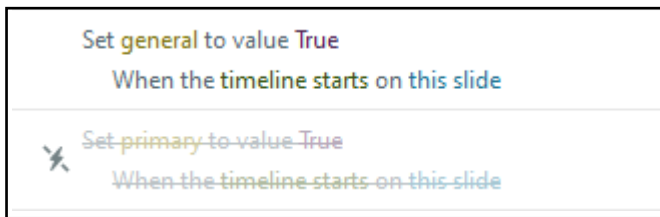


Navigation Functions

Variables and Triggers

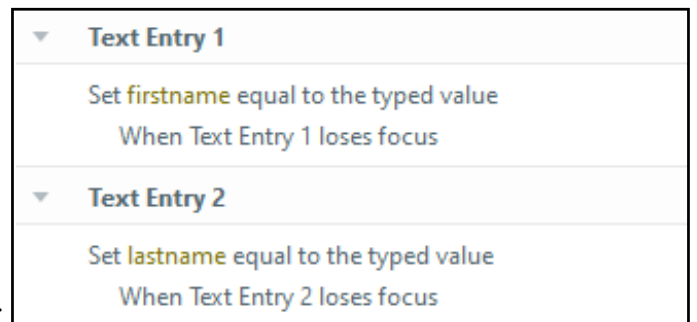
A programming variable is a data set that can hold information (i.e., a container for storing information). These variables are used frequently in traditional programming languages such as Python and Java but can also be used with limited functionality within Storyline as well. While most triggers in Storyline get reset once the user leaves their tied-to slide, variables are constants. Once a variable has been set to a value, it will remain unchanged until another trigger forces it to change. Storyline allows the user to create three different types of variables, all of which are used in the module:

- **True/False:** often known as a Boolean in programming, these variables can only exist as either true or false. This variable type is used in the module to change the version depending on what type of election is being conducted. In the below example, the “general” variable is set to “true” which then changes the state of slide elements (e.g., ballot style) to match the requirements of that specific type of election.



This allows our team to upload a new version of the module for each new election type with minimal maintenance or construction time required.

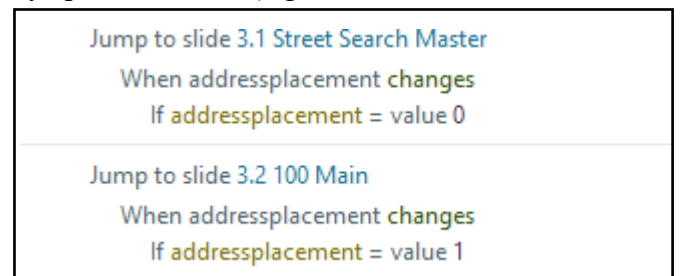
- **Text:** holds typed-in text values from the user. Used in conjunction with text-entry fields to capture and store



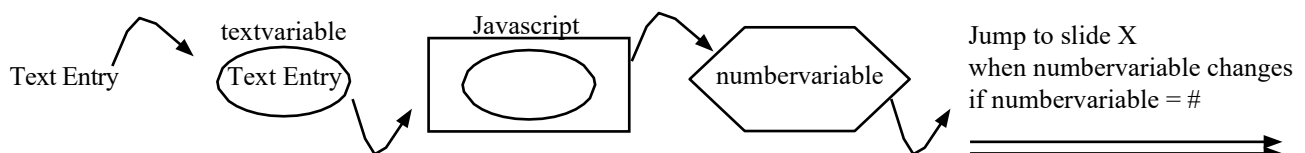
keyboard inputs from the user.

As seen in the pictures above and to the right, there are two text-entry fields on the Voter Search page. Text-based variables are then created for each one that transcribes the typed characters and captures them for later assessment. This functionality is the beginning of the search-emulation process that allows the user to move from page to page by typing information into the text-entry search bars.

- **Numbers:** holds numeric values that can be manipulated by specific events (e.g., “When timeline starts, add 1 to numbervariable”) or held in stasis and used as a conditional hypothesis for other functions. As you can see in the picture to the right, the slide the user will be moved to is based on the number stored in the variable when the variable changes. This functionality is the end-point of the search-emulation process that allows the user to move from page to page by typing information into the text-entry search bars.



As stated, the user begins by typing into the search entry boxes. Clicking the “View...” button or pressing the “Enter” key then begins a chain reaction:



Javascript

So then the question remains: how do we take the text the user is typing and transcribe it into a numbervariable that can be used to provide a slide trajectory for the user? We used triggers that would execute a Javascript when the textvariable value changed. That textvariable is then packaged up and plugged into the script as well.

As nobody on our team knew Javascript before creating this project, there was rigorous amounts of learning and testing when putting together a functional script that accurately emulated VAST.

```
1 //Street Search Declarations
2
3 var player = GetPlayer();
4
5 var finalnumber;
6
7 var houseparameters = player.GetVar("houenumber");
8 var houseLower = houseparameters.toLowerCase();
9 var housekey = "10 100 120 1209 250 2599 1214 1015";
10 var housetermlocation = housekey.search(houseLower);
11
12
13 var streetparameters = player.GetVar("streetname");
14 var streetLower = streetparameters.toLowerCase();
15 var streetkey = "mai main dub dubose glen abbey evans twelve oaks peace";
16 var streettermlocation = streetkey.search(streetLower);
17
18
19 var cityparameters = player.GetVar("city");
20 var cityLower = cityparameters.toLowerCase();
21 var citykey = "holly spring holly springs gar garner cary morrisville raleigh apex";
22 var citytermlocation = citykey.search(cityLower);
23
24
25 //Street Search Conditionals
26 if (housetermlocation === 3 && streettermlocation === 4 && citytermlocation === 0) {
27     finalnumber = 1 //100 Main
28 } else if (housetermlocation === 3 && streettermlocation === 4 && citytermlocation === 13) {
29     finalnumber = 2 //100 Main, Holly Springs
30 } else if (housetermlocation === 11 && streettermlocation === 13 && (citytermlocation === 0 || citytermlocation === 31)) {
31     finalnumber = 3 //1209 Dubose, Garner. User can type in city or leave it blank
32 } else if (housetermlocation === 16 && streettermlocation === 20 && (citytermlocation === 0 || citytermlocation === 38)) {
33     finalnumber = 4 //250 Glen Abbey, Cary. User can type in city or leave it blank
34 } else if (housetermlocation === 20 && streettermlocation === 31 && (citytermlocation === 0 || citytermlocation === 43)) {
35     finalnumber = 5 //2599 Evans, Morrisville. User can type in city or leave it blank
36 } else if (housetermlocation === 25 && streettermlocation === 37 && (citytermlocation === 0 || citytermlocation === 63)) {
37     finalnumber = 6 //1214 Twelve Oaks, Apex. User can type in city or leave it blank
38 } else if (housetermlocation === 30 && streettermlocation === 49 && (citytermlocation === 0 || citytermlocation === 55)) {
39     finalnumber = 7 //1015 Peace, Raleigh. User can type in city or leave it blank.
40 } else {
41     finalnumber = 0 //Street Search Master for any non-integrated street termlocation combinations. Will retain search bar contents.
42 }
43
44 player.SetVar("houseplacement",housetermlocation);
45 player.SetVar("streetplacement",streettermlocation);
46 player.SetVar("cityplacement",citytermlocation);
47
48
49 player.SetVar("addressplacement",finalnumber);
```

At its core, the script works to do seven things in order:

1. Establish a connection to the Storyline player module.
2. Declare “parameter” variables that capture and store the recorded textvariables from each text-entry field.
3. Transform the “parameter” variables into lowercase. This is required for the Javascript to function.
4. Provide a keystring for each text-entry field, listing accepted words. This is functionally equivalent to an online shopping search engine. If the keyword that the user is typing into an online retail website does not match with anything they have in their keystring inventory, then the user will not get any results.
5. Establish a new “termlocation” variable that searches for the “parameters” variable within the keystring to see if there is a match and then captures the results.
6. Use conditional “if...else if...else” strings to change the number of the “finalnumber” variable if certain number combination statements are true.
7. Set the numbervariable in the Storyline player module according to the “finalnumber” variable.

If you would like to learn more about how it works or view the entire code, two javascript files are also available: one that provides an in-depth explanation and one that contains the whole code.

Implementation

Testing and Finalizing

Once the exercise was finished, it was distributed to various teams within our office for review, including the IT team who was able to verify the accuracy of the emulation. This period involved rigorous feedback collection, stress-testing the module, and ensuring everything was technically correct.

Once we had made all necessary technical corrections and made the Javascript code as robust as possible, we uploaded the module to the web, unpacked the exercise packet, and uploaded everything onto the Training Resource page of our Precinct Official blog.

Spreading the Word

While doing final tests, we began marketing the new module to our Precinct Officials. A month before training, we introduced the *At-Home Provisional Exercise* in an episode of *Wake Up with Wake Elections*, an online video web-series we run to stay connected with our Officials and keep them apprised about upcoming changes. We were passionate about this improvement to our training and wanted to share that with people. During the live web screening, we received a lot of comments from Officials remarking about their excitement!

Once training began, we also allocated a 5-minute segment at the end of the lecture portion of our Help Table class to tell the Officials about the module, explain its benefits, structure, and how to access it. In nearly every class there were Officials making comments about the *At-Home Provisional Exercise* and how excited they were to be able to get more hands-on practice. Additionally, we recommended that our Chief Judges email their Help Table Officials a week or two before Election Day and remind them that the exercise exists and is a great resource for getting refreshed and ready for the big day!

Feedback

After every election, our team sends out a survey to our Precinct Officials. Our goal with this questionnaire is to gather feedback and see what areas we can improve upon in future elections; our team wants to know what went well and what didn't. As the *At-Home Provisional Exercise* was new, we wanted to gather feedback for that as well to see how it can be improved in the future. As a whole, we received thirty-three comments regarding the module, most of which were overwhelmingly positive!

Our survey also asks Officials to rate the module from "Poor" to "Excellent". As a whole, twenty-two commenters rated it "Excellent", seven rated it "Good" and four rated it "Fair". The constructive criticism we received revolved around multiple core things: a desire for the exercise to be printed off and handed to them during training, more exercises to be added, answer-sheet explanations, or for the module to be more interactive.

Very good practice. It would be helpful if it stated the exercises have to be printed out . Also ,
One case I had difficulty getting the correct answer. Who would we contact to discuss?

It helped having the exercises to work on, as they helped me be more confident with the help
table work.

I really liked the addition of this online module for help desk exercises! Nice job with both the
development and user experience.

It was REALLY helpful to do the help table exercises the day before election day. They were fresh
in my mind and I felt ready from the start.

I think the new online provisional exercises are an excellent idea! A lot of Officials seemed excited
about practicing more after class. I think it's helpful for them to practice when there's no
pressure and the focus is just on understanding.

EAC Clearinghouse Award Criteria

Innovation

- Historically, training exercises have always been limited to in-person training due to technological and licensing limitations. The *At-Home Provisional Exercise* provides an opportunity to help our Officials succeed on Election Day by giving them the ability to practice a complex process as many times as they want from the comfort of their own homes. This additional flexibility gives Officials the time and experience they need to process provisional voters with confidence and minimal mistakes.

Cost-effectiveness

- The *At-Home Provisional Exercise* is completely free for Precinct Officials to use. Additionally, this program was created using software our team already had licenses for. As such, it did not cost any money to create.

Replicability

- This module can be replicated by following the same design principles and steps that were used here: replacing the VAST images with images of any Election Day voter/street search software, using triggers to allow the user to navigate from slide to slide, and using Javascript to create a search engine that ties to text-fields. In practice, the images only need to be replaced with pictures of whatever software other counties use in their Election Day precincts. Our team has considered using this to create a module for our Early Voting Officials as well.

Sustainability

- Election-specific details that would usually require seasonal adjustments to the module have already been developed and tied to the states of objects on each slide. This allows our team to publish a new version for each election with the activation of a single trigger. This front-end development enables us to publish new versions for future elections without the time burden caused by constant updates.
- Using Master Slides will allow our team to update the Voter Search, Street Search, and Voter Record screens in bulk, saving time if our IT team ever makes updates to VAST.
- One of the goals of this project was to ensure it could be developed, expanded, and maintained by future team members. As such, during its development, its functions and programming were all documented. A development guide of the core module was created in Microsoft Word. Additionally, an explanation version of the Javascript was also created which provides details about each line of code and a 6-step guide on adding new scenarios to the module. This documentation will allow any team member to maintain and expand the module.

Meets Category Requirements

- The *At-Home Provisional Exercise* uses various tools and programs in combination to creatively solve multiple problems and improve our training. Providing a way for our Precinct Officials to practice processing provisional voters increases the accuracy, completeness, and precision by which they go through the provisional process on Election Day; our Officials now have a safe space to make first-time mistakes and learn.
- Having six different exercises for our Officials to work through will allow them to process a variety of situations on Election Day with confidence because they will feel familiar.

Improves Outcomes for Voters and Election Officials

- The *At-Home Provisional Exercise* allows our Precinct Officials to execute the Provisional process with increased confidence, improves the accuracy of following procedures, and helps ensure that every voter's ballot is handled with the care and consideration it deserves.

Conclusion

Future Improvements

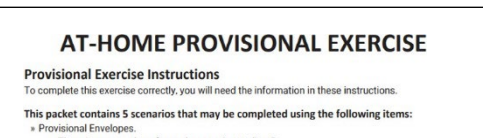
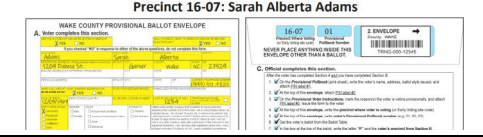
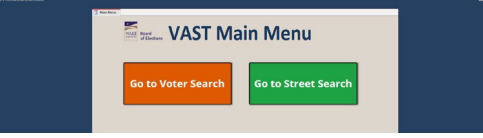
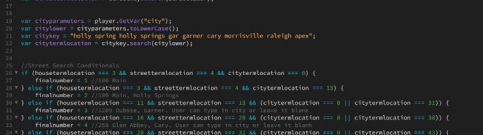
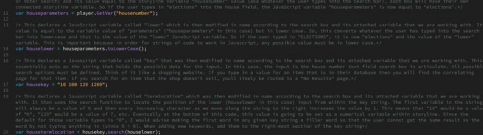

For future election we would like to expand the *At-Home Provisional Exercise* to offer a wider variety of possible scenarios. Although it currently offers five, each one different, we have received feedback from Officials for more! It could be beneficial to have multiple of the same type of scenario, putting a higher emphasis on scenarios that occur more frequently.

We would also like to improve and expand the provided answer sheet. As it currently stands, it simply provides answers to the fields that are required when completing each of these scenarios. However, it doesn't really explain why an answer is correct and why others could be wrong. As such, it is sufficient for checking your answers, but it does not help guide the user towards correct practices and clear up possible confusion points.

Conclusion

The creation and implementation of the *At-Home Provisional Exercise* has filled a long-time need in the Wake County Board of Elections training: it has allowed our Help Table Officials to receive a mental refresher on the provisional process and to feel comfortable and confident in processing Provisional Voters on Election Day. The response from our Precinct Officials has been overwhelmingly positive, and as we continue to build, expand, and impliment our innovative new tool as a core training resource, we hope to see a continued uptick in Precinct Official confidence and a decrease in procedural errors that will ensure every voter's ballot is considered with the care and accuracy it deserves.

Thank you for your consideration of the Wake County Board of Election's *At-Home Provisional Exercise* submission for the 2025 EAC Clearinghouse Awards in the category of Poll Workers: Best Practices in Recruiting, Retaining, and Training Poll Workers. Supporting documents can be found below.

<i>At-Home Provisional Exercise</i> - Resources	
 <p>AT-HOME PROVISIONAL EXERCISE</p> <p>Provisional Exercise Instructions</p> <p>To complete this exercise correctly, you will need the information in these instructions.</p> <p>This packet contains 5 scenarios that may be completed using the following items:</p> <ul style="list-style-type: none">Provisional Envelopes.	At-Home Provisional Exercise - Packet <i>This is the full packet that Officials print and fill out at home to practice processing provisional voters. Six different scenarios are included.</i>
 <p>Precinct 16-07: Sarah Alberta Adams</p> <p>WAKE COUNTY PROVISIONAL BALLOT ENVELOPE</p> <p>A. Make complete this section.</p> <p>B. Official completes this section.</p>	At-Home Provisional Exercise - Answer Sheet <i>Contains answer sheets for each of the required fields in the At-Home Provisional Exercise Packet. This allows Officials to check their work.</i>
 <p>VAST Main Menu</p> <p>Go to Voter Search</p> <p>Go to Street Search</p>	Voter and Street Tool (VAST) Web Simulation <i>Links to the web application our Precinct Officials used to simulate VAST and practice filling in the At-Home Provisional Exercise Packet</i>
 <pre>var cityParameters = {street: streetLocation}; var cityLower = cityParameters.toLowerCase(); var street = "fully capitalize street and give carter carterville calah walt"; var cityLocation = cityParameters.cityLower;</pre>	VAST Javascript Search Code <i>Contains the full VAST Javascript Search Code for Both Street Search and Voter Search. Also contains the script format that automatically capitalizes text-field entries.</i>
 <pre>var cityParameters = {street: streetLocation}; var cityLower = cityParameters.toLowerCase(); var street = "fully capitalize street and give carter carterville calah walt"; var cityLocation = cityParameters.cityLower;</pre>	VAST Javascript Search Code - Explanation Version <i>VAST Javascript Search Code with only core functions and built-in explanations for future team members. Explains with exact details and provides instructions for adding new scenarios.</i>
 <p>articulate Storyline 360 x64</p>	VAST Simulation Storyline File The original project file can be requested by emailing our team at elections@wake.gov .