



## **Spaceport Command and Control System Software Development**

Abraham Glasser  
NASA Kennedy Space Center  
Major: Computer Science  
Ground Systems Development and Operations  
Spring Session  
Date: 10 April 2017

# Spaceport Command and Control System Software Development

Abraham T. Glasser<sup>1</sup>

*Rochester Institute of Technology, Rochester, New York 14623*

The Spaceport Command and Control System (SCCS) is the National Aeronautics and Space Administration's (NASA) launch control system for the Orion capsule and Space Launch System, the next generation manned rocket currently in development. This large system requires a large amount of intensive testing that will properly measure the capabilities of the system. Automating the test procedures would save the project money from human labor costs, as well as making the testing process more efficient. Therefore, the ~~Electrical~~ Exploration ~~Systems~~ Division (formerly the Electrical Engineering Division) at Kennedy Space Center (KSC) has recruited interns for the past two years to work alongside full-time engineers to develop these automated tests, as well as innovate upon the current automation process.

## Nomenclature

GUI	= Graphical User Interface
CSV	= Comma Separated Values
<del>DA</del>	= <del>Development Activity</del>
<del>IDE</del>	= <del>Integrated Development Environment</del>
KSC	= NASA Kennedy Space Center
LCC	= Launch Control Center
LCS	= Launch Control <del>Sub</del> system
<del>MPEG</del>	= <del>Motion Picture Experts Group (video file type)</del>
<del>NACK</del>	= <del>Negative Acknowledged Response</del>
NASA	= National Aeronautics and Space Administration
NE- <del>XES</del>	= <del>KSC Engineering – Exploration Division/Software Branch</del>
OSB II	= Operations Support Building II
<del>SCCS</del>	= Spaceport Command and Control System
<del>SLS</del>	= <del>Space Launch System</del>
OCR	= Optical Character Recognition

## I. Introduction

The work ~~that was~~ done ~~in the duration of~~during this internship aimed ~~eds~~ to ~~develop automated software testing to~~ make the software development process more optimized by reducing the amount of manual testing for software ~~through developing automated software testing~~. ~~This software will be used for future Human Spaceflight Program launches~~. Making automated software tests will allow software engineers to spend their skills and time on other aspects of the project rather than spend a lot of their time manually testing software. Automating software tests will ~~also~~ help reduce human error that would appear, especially after hours of rigorous testing. After the automation process has been ~~practiced and~~ implemented ~~and refined~~ for this project, it ~~will apply~~ ~~can be applied~~ to future projects.

The majority of this development involves using an automated testing framework and an automation library. The automated testing framework has its own syntax and utilizes keywords to operate automation features. It ~~is implemented with Python and~~ automates anything that appears on a desired screen with the use of image recognition software to detect and control GUI components.

---

<sup>1</sup>Software Development Intern, NE-XS, NASA KSC, Rochester Institute of Technology

There are currently spreadsheets of test procedures that contain step-by-step test cases that would normally be followed by engineers. Generally, the tests we've automated test the given sections of a system GUI and produce measurements and statuses, which are then verified by checking the values that they indicate.

When we, the interns, could not work on our automated testing development in the LCC due to set scheduling conflicts or reprovisioning of the system, we would either attempt to blind code the automated test procedures to the best of our abilities or work on side projects that were individually assigned to the interns. These side projects mostly contained working on the current tools that assist the automation, as well as trying to incorporate more tools to make the automation process more efficient.

## II. Approach

To work on this project, I had to learn the objectives and tools by researching into each topic. Then, I ran tests, learning from trial and error until I developed a solid understanding of how to use the tools and software.

### A. Training/Familiarization

I started this internship in mid-January of 2017. For the first week, all of the interns that were working this semester were together, regardless if they were new or returning/yearlong interns. We introduced ourselves and met our intern coordinator and other important points of contact. We went through a few training sessions. These sessions were about security, safety, and other such topics. The second week we split up into our groups of interns with whom we would be working closely. In our group, there were three returning/yearlong interns and three of us new interns. We paired up and we watched each other work and became familiar with the system, where the files are stored, how to use version control software, etc. We watched the “senior” interns work during the second week, and then we got our own assignments and went straight to work, asking them for advice/assistance along the way.

Later on, after a couple of months, we ~~had~~ separated and categorized the general keywords normally used for our test cases based on to what the keywords are commonly applied. This change took a little adjustment for the team, because it changes the file paths that we reference to in our code, and causes keywords to “break”. However, ~~but for the in the~~ long term it will provide a more efficient process of searching for desired keywords. Because of this and other changes, including syntactical changes to the coding standards, we had to refactor what we had done and change the way we would develop future tests.

### B. Automated Testing for SCCS Dashboard

The test case spreadsheets we had been working from were written by system test engineers, people who manually tested the software and wrote guidelines for testing the software. Below is an example of what the major parts look like. The number of test ~~eases~~ steps in each case varies, but is usually around 60-120, ranging from 20 to 300.

Step #	Operator Action	Expected Response
18	Launch display, “Some Display”	“Some Display” display appears
19	Execute the command script “Some Script”, to Load data	Values on “Some Display” are changing

Table 1: Generalized Examples of Test Steps from a Test Procedure Spreadsheet.

Since I am not familiar with all the different displays in the software (and the possibility of them changing), I did some of the steps manually to familiarize myself with them before going ahead and coding the automation.

#### a. Technical Specifics

While writing test steps that will automate a GUI, there are buttons, text fields, editable value fields, and dropdown menus that need to be considered. This requires precise selection of those desired areas on the screen. Because we are not using the source code of the GUI to locate and select this items, we have to take pictures of the text we want to select or the button we want to click on. This involves taking a lot of screenshots and using image recognition tools to correctly find the matching images.

### *b. Test File Formatting*

The actual files that we write are high-level and are used by the automation software. They start with documentation and file paths to the resources that they use, as every test case is different. The majority of our work goes into writing the test steps, under the “Test Cases” section. The work we do is usually done in a general-purpose text editor named gedit. However, we could use any text editor, including those that are built into the operating systems’ command line interface. Below is an example of what the final result would look like. The number of test steps written usually matches the number of test steps in the spreadsheets.

```

1 *** Settings ***
2 #author: Andrew Hwang
3
4 Documentation      Sample Test
5 Test Setup         Set Libraries
6 Resource           Some Common File
7 Library            Some Library
8
9 *** Variables ***
10 ${SomeVariable} = Value
11
12 *** Test Cases ***
13 Sample Test Case
14     Run Keyword
15
16 *** Keywords ***
17 Run Keyword
18     Running Sample Keyword

```

Figure 1: Format of Sample Automated Test File.  
(Picture owned by Andrew Hwang)

## **C. Future Developments**

As of now, the automated tests rely on the image recognition capabilities and the images that we produced and captured, but we are getting closer to using a more effective method of OCR. This will only require screen shots of the entire screen, which is far more automatable than specifically capturing portions of the screen and creating images whenever something on the screen is updated. This will also make the automated test cases more adaptable to a change in the screen’s environment. We have been working on getting these tools incorporated in our systems.

## **III. Conclusion**

So far we have finished about 60% of all the automated testing there is to do. This 60% consists of tests that we have already run in the LCC to make sure they work. However, we have been delayed at times due to not having access to the systems in the LCC to run and make sure our tests are working. When we experienced this, we just kept on “blind coding” the test cases, writing them based on our knowledge of the displays and such. This does help us make progress because later when we get access to the computers in the LCC, we will only have to make a few changes to get the test cases working. The work we are doing is important, and we used the actual software and hardware that is going to be used to launch NASA’s Space Launch System, the next generation of rockets destined to go to Mars. Our work has proven itself cost and time-effective.

### **Acknowledgements**

I would first like to thank Caylyne Shelton for reaching out to me regarding my application and helping me get all the information I needed for this internship. She helped me greatly before and during the internship. Anytime I had any questions about KSC, my internship, absences, etc. I never hesitated to ask her. I don't just see Caylyne as my mentor/supervisor, I see her as my friend, she has been very helpful making sure I am doing okay and have everything I need. I would also like to thank Jamie Szafran for being there when Caylyne became tremendously ill. Jamie has done a great job answering our questions, giving us advice, and helping us have fun. While Caylyne and Jamie heard us out and helped us get what we need, Oscar Brooks, the branch chief, worked behind the scenes granting us permission and access to everything. He has a lot of work to do, and to find the time to help provide for us interns means a lot.

Coming to work every day wouldn't be as fun without the other interns on my team, Andrew Hwang, Michael Backus, Mark Rodriguez, Meriel Stein, and Tom Plano. They have been great friends throughout this internship and I'm sure they will continue to be so after the internship finishes. Andrew, Michael, and Mark are experienced interns, and I would like to thank them for helping us new interns learn and progress with our tasks.

I am deaf myself, and I needed a sign language interpreter and a video phone for my internship. Eddie Wroblinski, the Equal Opportunity Specialist here at NASA's KSC, reached out to me some time before the start of my internship. He is very friendly and made me feel welcome coming to KSC. He made sure I had an interpreter and asked me if I needed anything else while I was here. A couple of weeks into the internship, he took the time to come into my office in OSB II and check on me and my interpreter to make sure everything was all set. I would also like to thank Jessica Conner, the Equal Employment Opportunity Specialist here at KSC, for helping set up my video phone in my office in OSB II. She is deaf herself, and she made sure everything was ready for me.

I would especially like to thank Cindy Spann, one of the full-time sign language interpreters here at KSC. Cindy has been my primary interpreter while I have been interning here. She has been my interpreter almost every day, and she has been a huge help. She is very friendly and warm hearted, she and the other interns in my office got along so well, and she helped interpret everything they said, and the interns didn't feel uncomfortable with her here at all. Cindy also helped me get to know KSC, showing me and telling me stories about people, the buildings, and even weather conditions here at KSC. She also helped me outside of my internship, giving me advice about a variety of things from attractions to hospitals in Florida. She has been a great interpreter and all of us interns look forward to seeing her every day here at work.