SAS® GLOBAL FORUM 2020 #SASGF

Paper SAS4517-2020

Look Ma, No Hands: How to Train Your Dragon for SAS®

Julianna Langston and Elizabeth Langston, SAS Institute Inc.

ABSTRACT

Have you ever experienced pain in your hands while using a keyboard? Have you wondered how to write SAS® code without using your hands? Well, you will love this paper. After a SAS accessibility developer received a repetitive strain injury (RSI) diagnosis, she started using Dragon® NaturallySpeaking (a speech-to-text product) to write SAS code hands-free. In this presentation, she describes tips for getting started with Dragon, demonstrates techniques for training Dragon to understand SAS language vocabulary, and shares a case study for using Dragon with her favorite SAS programming client.

INTRODUCTION

As a software developer at SAS, I am part of a development team that creates an accessibility product and helps other teams improve the accessibility of their products. Our job is to give SAS customers of all abilities a better experience with SAS software.

Over the past decade, I'd spent most of my workdays writing code, with my hands on a keyboard. In summer 2019, I began to notice tingling in my hands after hours of typing—just a few twinges. They were annoying but did not affect my productivity. Over the next several weeks, the soreness worsened, but the discomfort was so minor, I did not even think of it as "working through" the pain.

Then one day, the pain exploded. I had to back away from the computer, no longer able to use my hands. My first thought, obviously, was "How do I hide this? No one can know." In an instant, my commitment to "pretending I do not have an injury" went from a passive hobby to an active mission. I scheduled more tactical meetings to avoid typing, focused on projects that relied less on typing, and sought help from a doctor (who did not use the magic fix-everything wand like I specifically asked for).

Typing ten hours a day for fifteen years had apparently negatively impacted my health. I was diagnosed with a Repetitive Strain Injury (RSI) and, until my hands healed, I would have to find alternatives to my dependence on a keyboard. In the meantime, SAS provided me a license of Dragon NaturallySpeaking.

Dragon NaturallySpeaking is a speech-to-text product that enables a user's spoken words to be converted to text in four steps:

1. The user speaks content into a microphone.
2. Dragon interprets the user's speech and converts it to text or keyboard commands.
3. A close approximation of the user's words is written to a file.
4. If necessary, the user revises the text to its intended content.

Dragon seems to work best for users who want to dictate paragraphs into documents. My usage—writing and editing syntax-heavy software code—felt like a fringe use case. There was a definite learning curve, both personal and technical. I will share both my challenges and successes so that you can avoid my mistakes.

PERSONAL LESSONS WITH DRAGON NATURALLYSPEAKING

The following are lessons that I learned while working with Dragon NaturallySpeaking:

- *Unrealistic expectations*: It frustrated me that my productivity was lower with Dragon than with healthy hands. I had to remember that "thinking with my hands" is always faster than thinking out loud. Writing code might be less efficient with Dragon, but it is far more effective than working in pain or not working at all.
- *Temptation to cheat*: I trained on Dragon the old-fashioned way: dig through documentation just enough to get started, and then wait until I hit an obstacle before digging again. Since I am able to use the keyboard for brief periods, I sometimes solved obstacles with my hands. "Cheating" meant that I did not learn Dragon as well as I needed and that I re-injured my hands.
- *Focused training*: Eventually, I discovered that cutting corners on the learning curve was not working out. I paused my assigned projects for a few days and focused on learning Dragon. Using a SAS program that I'd already written, I learned the tool by re-creating familiar code through speech-to-text. The process taught me a lot about Dragon and uncovered plenty of gotchas to resolve.
- *Being self-conscious about my speech patterns*: The mental process of composing prose out loud was surprisingly difficult. If I am in a conversation with another person, I can be comfortable and informal with my diction. I do not have that type of freedom with Dragon. With the tool interpreting each word as I spoke it, I had to become more aware of what I said and how I said it. Early on, this made me feel uncomfortable and awkward while simply using the tool.

TECHNICAL RECOMMENDATIONS FOR DRAGON NATURALLYSPEAKING WITH ANY SOFTWARE

Once you have gained some familiarity with Dragon, here are four suggestions for using this tool with any software product, such as desktop applications and web browsers:

- *Have a paper copy of the documentation available.* I printed a paper copy of the Dragon features that I used most often, for example, Chrome commands and desktop navigation. I keep it within reach as a reference.
- *Become comfortable with the Dictation Box.* It is critical to be well versed with the dictation box. You will use it often, especially as a rich editing space.
- *Install the Chrome extension for Dragon NaturallySpeaking.* Using the Chrome extension provides some nice, added features, such as rich editing inside the browser or jumping straight to a particular link or edit field. Internet users have a better experience with the Chrome extension installed.
- *Be proactive about vocabulary.* You will have to "train your Dragon" to understand what you mean. Consider making a list of the words, acronyms, and phrases that you use most often with your IDE and programming languages. Dragon might misinterpret what you mean ("sass" instead of SAS) or overload a word with their own meaning. Add your list as vocabulary and keep it updated as you discover other words to include.

TECHNICAL RECOMMENDATIONS FOR USING DRAGON NATURALLY SPEAKING WITH SAS SOFTWARE

For SAS programmers, here are recommendations that are specific to using Dragon NaturallySpeaking with the SAS Language and SAS clients:

- *Become fluent with the Dictation Box.* You will use the Dragon dictation box for composing and rich-editing of code in all SAS clients (for example, SAS® Studio, SAS® Enterprise Guide®, and SAS® Display Manager). It is essential to invest time learning the dictation box.

- *Become fluent with SAS keyboard commands and shortcuts.* Learn the keyboard commands and shortcuts that are available in your SAS client. This information improves your ability to navigate the clients with Dragon.
- *Familiarize yourself with MouseGrid.* Not everyone is a fan of MouseGrid, but it is a good idea to familiarize yourself with it to see whether it works for you.
- *Use existing SAS code to learn Dragon.* You have two skills to acquire for using Dragon with a SAS client: how to dictate so that Dragon interprets your intentions correctly and the mindset of composing new code verbally. It might be easier to practice these two skills separately.

Try dictation with existing SAS programs first. Speak the keywords, member names, and syntax of a familiar code fragment—noting where Dragon interprets well and where you will have to adjust. For example, "select" is a SAS keyword used often with the SQL procedure. Dragon interprets "select" as a keyboard or mouse command. See the Appendices for additional tips and examples.

CONCLUSION

When circumstances prevent you from using a computer keyboard, Dragon NaturallySpeaking provides a solid alternative. It has a significant learning curve, whether you are using desktop applications (for example, Microsoft Word), web browsers, or SAS. Once you have invested time in learning and adapting Dragon to your speech patterns and job requirements, you will have a tool that gives you an effective option for getting your work done.

RECOMMENDED READING

SAS client keyboard shortcuts:

- *Keyboard shortcuts in SAS Studio 5.2*
- *PDF cheat sheet alternative*
- *Enterprise Guide keyboard shortcuts*
- *Base SAS (also known as DMS) keyboard shortcuts*

SAS client keyboard shortcuts:

- *Dictating text*
- *Web browser basics*
- *Working with Chrome*
- *Working on your Desktop (Windows)*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Julianna Langston
Julianna.Langston@sas.com

APPENDIX A: USEFUL DRAGON KEYWORDS WITH SAS LANGUAGE AND CLIENTS

Dragon Keyword	Usage
Backspace	The equivalent of pressing the backspace key.
Cap	Capitalizes the next word.
Caps on	Force all characters to be uppercase. This can be turned off by saying “Caps off”.
Comma	Adds a comma.
Dot	Adds a period, or “.” This is different from saying “period”, which adds a dot and a space: “. ”
Equal sign	Adds a “=”
New line	The equivalent of pressing the enter key.
No caps	Do not capitalize the next word.
No caps on	Force all characters to be lowercase. This can be turned off by saying “No caps off”.
No space	If you want to say two words, but you do not want a space between the words (like “noprint”), say the first word, then “no space”, then the second word. You might need to pause after the first word.
Numeral	If you want to add a single number, you must say “numeral”, then the number. So, “3” is said as “numeral 3”. If you only say “3”, it prints as “three”. If you `want to add a larger number, like “12” or “400”, you can just say “twelve” or “four hundred”, without the “numeral” in front.
Open paren / close paren	Adds “(” or “)”
Open quote / close quote	Adds double quotation marks. In addition, if you are in the dictation box, you can say a phrase, like “This is a description.” After you see Dragon print that text, say “quote that”, and quotation marks will be added around whatever phrase you just said.
Period	Adds a dot and a space: “. ”
Press _	Presses whatever key you say. So, “press F5” or “press escape” or “press J” presses those keys. Note: sometimes when I try to say “press n”, instead of printing “n”, Dragon prints “press in”. In those cases, I have to say “press November”, and that is when Dragon prints only “n”. “November” is the word in the military alphabet for “n”.

Dragon Keyword	Usage
Select	Selects
Semi-colon	Adds a ";"
Spacebar	The equivalent of pressing the spacebar.

APPENDIX B: ADDITIONAL TIPS FOR USING DRAGON NATURALLYSPEAKING

- Add vocabulary. When I first started using Dragon, whenever I said "SAS", Dragon printed "sass". I could just say "S A S", and that would accomplish the same goal, but it makes it faster to just add "SAS" to the vocabulary list.
- Printing single word Dragon commands. There are many words that Dragon interprets as commands or punctuation marks, for example, "select" or "period". If you want Dragon to actually print that word, instead of using it as a command, you have to trick it. To get Dragon to print "select", I say "selected". After Dragon prints "selected", I say "backspace 2". Dragon hits the backspace key twice, and I ultimately have the word "select". To get the word "period", I say "periodic [pause] backspace 2". To get the word "comma", I say "command [pause] backspace 2".
- Printing multi-word Dragon commands. There are several commands that use multiple words, for example, "no space", "move up", and "select next line". Dragon is listening for the full command. So, if you say only part of the phrase, Dragon does not interpret it as a command. If I say "move", and then say "up", Dragon ends up printing the phrase "move up", instead of actually moving the cursor up one line.
- Managing capitalization. SAS is a generally case insensitive language, so managing your capitalization might not be strictly necessary. However, there are typically style guides that people follow and want to continue to follow. For example, if I want to print the phrase "sashelp", I say "SAS no space help". The word that Dragon prints for me, though, is "SAShelp". This still works as a LIBNAME, but is not how I have always printed it. In order to get the LIBNAME to look the way I wanted to, I say "no caps on SAS no space help no caps off".
- Selecting lines. When my cursor is on a line of text, be it in SAS code or in a document of prose that I am editing, I might want to edit a block of text within the dictation box. If I can highlight some text, and then say "open dictation box", whatever I highlighted is opened for editing. In order to use Dragon to highlight lines of text, I can say "select next line", "select next five lines", "select previous line", "select next five words", and so on.
- **Resorting to "press"**. Sometimes, no matter what I do, Dragon really wants to add spaces and capitalization where I type. I wind up hitting this most often when I am trying to print dates using the date. format. For example, if I want to print "01Jan10", I cannot get Dragon to consistently type it out without adding spaces after the fact. I should be able to say "numeral zero no space numeral one no space jan no space 10", but sometimes spaces get added after the fact. When I get frustrated enough with Dragon that I cannot get it to type exactly what I want, I just resort to using the word "press". If I say "press zero", Dragon prints "0".

You can try out some of these tips in appendix C, where I show you scripts that you can read to Dragon to get Dragon to print SAS code. This includes some of the cases that I had to learn to work around as I started using Dragon as a SAS programmer.

APPENDIX C: EXAMPLES OF COMPOSING SAS CODE WITH DRAGON

The tables below display five examples of SAS PROC steps.

- Column 1 demonstrates precisely what a user would have to speak in order to compose the required PROC step. The Dragon-specific commands are underlined. See Appendix A for more information about each command.
- Column 2 displays what the resulting code would be after Dragon interprets the user's speech. Note that the exact capitalization might be different from how a programmer might normally type it.

Example 1: Use the PRINT procedure to display the specified columns in SASHELP.CARS.

Speech to Dragon	Resulting Code

proc print data equal sign S A S no space help. dot cars semicolon new line
var model type M S R P semicolon new line
format M S R P dollar. no space. numeral eight. dot semicolon new line
run semicolon
	Proc print data = SAShelp.cars;
	var model type MSRP;
	format MSRP dollar8.;
	run;

Example 2: Use PROC SQL to display all columns and all rows of SASHELP.CARS.

Speech to Dragon	Resulting Code

proc S Q L semicolon new line
selected. backspace 2. spacebar star spacebar from S A S no space help. dot class semicolon new line
quit semicolon new line
	Proc SQL;
	select * from SAShelp.class;
	quit;

Example 3: Use PROC SQL to derive a new table from SASHELP.PRDSAL2.

Speech to Dragon	Resulting Code

Proc S Q L no no space print semicolon new line
Create table my no space cap table as new line
Selected. Backspace 2. Distinct product comma product backspace 3 no space type new line
From no caps on S A S no space help dot P. R. D. S. A. L. no space numeral 2. New line
Quit semicolon
	Proc SQL noprint;
	create table myTable as
	select distinct product, prodtype
	from sashelp.prdsal2;
	quit;

Example 4: Use the MEANS procedure to display basic statistics for SASHELP.CARS.

Speech to Dragon	Resulting Code

Proc means data equal sign S A S no space help dot cars min median max S T D range spacebar press N semicolon new line
By make semicolon new line
Var engine no space size M S R P semicolon new line
Run semicolon
	Proc means data = SAShelp.cars min median max STD range n;
	by make;
	var enginesize MSRP;
	run;

Example 5: Use the SGPLOT procedure to display a graph from SASHELP.STOCKS.

Speech to Dragon	Resulting Code

Proc S G no space plot data equal sign S A S no space help dot stocks open paren where equal sign open paren date greater than sign open quote press 0 press 1 press j press a press n press 0 press 0 close quote press d close paren close paren description equal sign open quote stocks after 2000 close quote semicolon new line
Series x equal sign date y equal sign adjusted. Backspace 5. No space close spacebar slash spacebar group equal sign stock semicolon new line
Run semicolon
	Proc SGplot data = SAShelp.stocks (where = (date >"01jan00"d)) description = "stocks after 2000";
	series X = date Y = adjclose / group = stock;
	run;