Getting Started with Stock Synthesis (SS3)

SS3 Development Team

Last Updated: February 16, 2022

1 Scope

Stock Synthesis (SS3) is a statistical fisheries population dynamics modeling framework built in AD Model Builder. This Getting Started Guide introduces users to running SS3. We assume some experience with population dynamics models and a basic understanding of how to use the command line.

If you have never used the command line, please see the resources in Appendix 1.

To follow along, we recommend downloading the simple example. Simple is a working SS3 model that runs in less than three minutes and should allow you to experiment with SS3 workflows without having to worry about the model's contents.

By the end of using this guide, you should:

1. Understand the input and output file structure for SS3
2. Be able to run an SS3 model via the command line
3. Understand which tools are available to work with SS3 models
4. Troubleshoot basic problems associated with running an SS3 model

2 Stock Synthesis file structures and tools

SS3 uses text input files and produces text output files. In this section, the SS3 input and output files are described, as are tools that can be used with these files. See how they relate in Figure 1.

2.1 SS3 files: Required inputs

Four required input files are read by the SS3 executable. Throughout this document, we will refer to the SS3 executable as ss.exe. Keep in mind that the Linux and Mac versions of SS3 have no file extension (e.g., ss), and the executable can have a different name (e.g., ss_win.exe, ss_3.30.18.exe). These input files are:

1. **starter.ss:** Required file containing file names of the data file and the control file plus other run controls. Must be named starter.ss.
2. **data file:** File containing model dimensions and the data. The data file can have any name, as specified in the starter file, but typically ends in .ss or .dat.
3. **control file:** File containing set-up for the parameters. The control file can have any name, as specified in the starter file, but typically ends in .ss or .ctl.
4. **forecast.ss:** File containing specifications for reference points and forecasts. Must be named forecast.ss.

Figure 1: SS3 input and output files, and associated tools. Note that some tools are used with both input and output model files.

2.2 SS3 files: Optional inputs

The two optional input files for SS3 are:

1. **ss.par:** A text file with one line per parameter where order matters. Usually created from a previous model run. This file can provide initial parameter values instead of using the initial values in the control file. The option to use the ss.par file or the control file for initial parameter values is selected in starter.ss.
2. **wtatage.ss:** File containing empirical input of body weight by fleet and population and empirical fecundity-at-age. The option to use the wtatage.ss file is selected in a line of the control file.

2.3 SS3 output files

Many output text files are created during a model run. The most useful output files can be divided into:

1. **Files containing results**. These files include ss_summary.sso and report.sso that contain model results summarized in different ways.
2. **.ss_new files**. These files echo the SS3 input files, but include standardized comments. The values should be the same as the input files, except for control.ss_new, which updates the initial values to the final estimated values. The .ss_new files can be useful to standardize comments in the input files and can be checked to make sure that SS3 interpreted the inputs as the user intended.
3. **Files used for debugging**. These files include warnings.sso and echoinput.sso.

2.4 Tool definitions

- **Any text editor** can be used to view and edit SS3 input and output files. Defining a custom syntax highlighting scheme for SS3 files may improve SS3 file readability. Some commonly used text editors include Atom, Emacs, Notepad++, and Sublime Text.

- **r4ss**: An R package to plot SS3 model results and manipulate SS3 input and output files. Available at: https://github.com/r4ss/r4ss

- **SSI**: Stock Synthesis Interface (i.e., the SS3 GUI). The latest version can be downloaded from within the document library on Vlab. SSI can be used to edit, save, run, and visualize model inputs and outputs.

3 Running SS3

SS3 is typically run through the command line. We will introduce three common command-line approaches:

1. The one folder approach
2. The two folder approach
3. The path approach

Other possible approaches to running SS3 include:

1. Run SS3 from within R (discussed briefly)
2. Use SSI to set up model files and run (not discussed here; find the SSI and its documentation

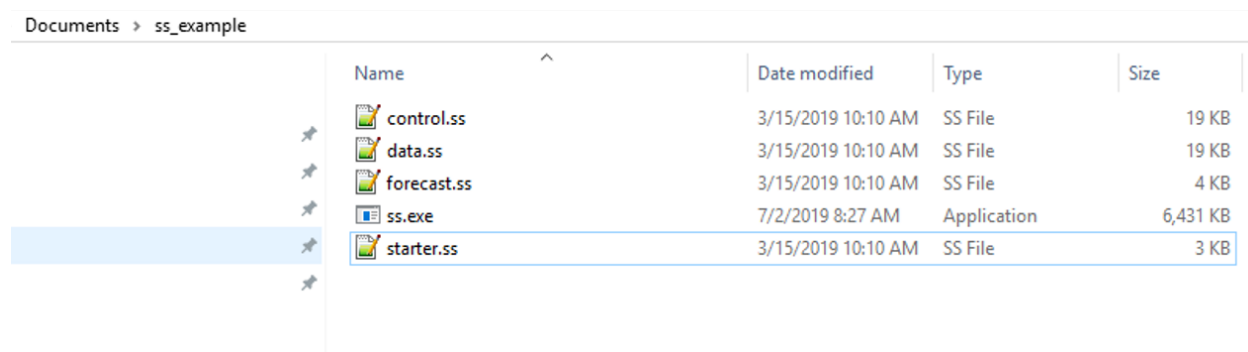## 3.1 The one folder approach and demonstration of an SS3 model run

The one folder approach is so named because the model input files and SS3 executable are in the same folder. The one folder approach is the simplest way to run SS3 via the command line, so we will walk through an example run using this approach.

3.1.1 Setup for the 1 folder approach

Create a folder and add:

- Control File (Must match name in starter.ss)
- Data File (Must match name in starter.ss)
- forecast.ss
- ss.exe (or ss_opt.exe for running faster without internal checks)
- starter.ss
- Conditional files: wtatage.ss (if doing empirical wt-at-age approach) and/or ss.par (to continue from a previous run)

For example, here is what should be included for a model with no conditional files (where the control and data file names specified in starter.ss are control.ss and data.ss, respectively):

3.1.2 Run SS3

Once all of the model files and the SS3 executable are in the same folder, you can open your command window of choice at the location of the model files.

To do this, you can typically click to highlight the folder the model files are in, then shift + right click on the same folder and select the option from the menu to open the command line of choice (e.g., Windows Powershell). This should bring up a command window. Then, type `ss` (or other name of the ss exe) into the command prompt and hit enter. Note that if you are using Windows Powershell, you will need to type `./ss`.

The exact instructions for running SS3 can differ depending on the command window used. If you have trouble, search for resources that describe running an executable for your specific command line.

If you still have issues, double-check that you are calling the correct exe name (e.g., you would need to type `ss_win` to run an executable called `ss_win.exe`.) and that the command line has the correct directory.

After starting the run, you should see output to the command line similar to:

```
Windows PowerShell                                                    —    □    ×
PS C:\Documents\ss_example> ./ss
Error trying to open data input file C:\Documents\ss_example\ss.dat
 reading from STARTER.SS
 reading from data file
Data read sucessful 999

 reading forecast file
 reading from control file
If you see 999, we got to the end of the control file successfully! 999
 end of parameter section
 ready to evaluate once in prelim

Estimating...please wait...
0 0 -log(L): 0   between
 MG setup OK
 growth1 OK
 natmort OK
 migr OK
 selex OK, ready to call ALK and fishselex
 ready for virgin age struc
 ready for initial age struc
 OK with initial conditions
 OK with time series
 did survey obj_fun   0 20.9926 18.6847
 did lencomp obj_fun    600.838 173.276 0
 did agecomp obj_fun   526.434 170.079 0
 did meanlength obj_fun   39.4 41.6308 0
 initequ_catch -log(L) 0
 catch -log(L)  0.00800702 0 0
 did recruitdev obj_fun -15.8356 31 0.72
 OK with obj_func 1576.56
```

If the output gets past `Estimating...please wait...`, then the structure of the SS3 inputs is OK. The volume of information displayed during the run within the command window is controlled in starter.ss file on the line with the comment `# run display detail (0,1,2)` (1 is the most typically used value).

SS3 always opens and reads files in the same order (starter.ss, data file, forecast.ss, control, then wtatage.ss if using, and finally ss.par if using), writing to echoinput.sso as it reads. SS3 inputs are read in order by value, without reference to comments (i.e., anything after `#` on a line). Depending on which options are selected, SS3 will expect a particular number of values in a certain order, so failing to provide them will result in errors or at least a different model specification than intended.

After reading, SS3 proceeds immediately to pre-processing the data and creating internal parameter labels. As SS3 executes, it writes checks to echoinput.sso and warnings to warnings.sso.

SS3 next goes to the procedure section where iterative parameter changes are made by ADMB to minimize the negative log likelihood. When ADMB achieves convergence, control passes to the sd_phase for calculation of parameter variance (unless `-nohess` option is specified; See command line options). After the sd_phase, SS3 proceeds to the benchmark and forecast code section and then to final reporting. Output files containing results are written to the directory.

When the run completes, SS3 shows `!! Run has completed !!` and the number of warnings:

Examine warnings.sso for issues and suggestions, regardless of if the run completed successfully or not. If SS3 exits on error, you will not see the message !! `Run has completed` !!. Here is an example of a run that exited on error due to invalid inputs:

3.1.3 Examine the output

.ss_new files are generated if a model run completes. For more information on the .ss_new files, see the "Output Files" section of the SS3 user manual.

ss_summary.sso files are included with the examples on vlab so that you can compare it with the `ss_summary.sso` file generated by your model run. You should see similar if not exactly the same values for the likelihood and parameter values if using the same version of SS3.

Output from SS3 can be read into r4ss or the excel viewer.

3.2 The two folder approach

With this approach, you can have one (or more) folders with various versions of SS3 and many model user folders each with one (or more) batch file(s) that point to various versions of SS3. This eliminates the need

to have multiple copies of the same SS3 executable on your computer, which is necessary with the two folder approach.

In this approach, the ss.exe is no longer in the same folder as the model; instead, a batch file with a path to the SS3 executable is included in the folder with the model file.

The example shown in this handout uses batch files, which can be used on Windows only. However, there are similar workflows that can be used on OSX and Linux.

3.2.1 Example

Put the model files in one folder and the desired executable in the other:

Within the ss_exes folder, any number of SS3 exe versions could be added, but for now, we only have one:

The other folder contains the model files, but also a batch file, which is a text file with the extension .bat:

The batch file, a text file containing commands, can have any name, but must end in the extension .bat. In this case, we have called the batch file run_ss.bat. The first line of the batch file has the full path to the SS3 executable. For example, our batch file contains only the line:

Open the command window in the model user folder (in this example, the ss_example folder), then type

the name of the batch file (e.g., in this case, type `run_ss` or `./run_ss`). This will call the batch file, which will then call the SS3 executable to run in the folder with the model files. If the model runs, you will see output to the console as with the one folder approach.

This basic approach using a batch file can also be used to do multiple model runs, all called from the same batch file.

3.3 The PATH approach

The PATH is where your computer searches for files by default. This means that if an SS3 executable called ss.exe is in your path, when you type `ss` into the command window, regardless of the directory, your computer will be able to find ss.exe and use it with the model files in the current directory.

For more information on how to add Stock Synthesis to the computer's PATH, see Appendix 2.

3.4 Run SS3 from within R

It is possible to run SS3 from within the R console. When the executable ss.exe in the same folder as model input files, use `system("ss")` from the R console to run. When using ss.exe that is in your PATH in Windows, use:

```
get_bin <- Sys.which("ss.exe")[[1]] # get ss exe that is in your path
system(get_bin) # to run SS3 in current working directory
```

or the function within r4ss:

```
# run 2 models, in directories folder_1 and folder_2, using the SS3 executable
# named ss that is in the path.
r4ss::run_SS_models(dirvec = c("folder_1", "folder_2"),
                    model = "ss",
                    exe_in_path = TRUE)
```

Running SS3 from within R may be desirable for setting up simulations where many runs of SS3 models are required (e.g., ss3sim) or if r4ss is being used to read model output.

4 Options for running SS3

4.1 Command line options

ADMB options can be added to the run when calling the SS3 executable from the command line. The most commonly used option is `ss -nohess` to skip standard errors (for quicker results or to get Report.sso if the hessian does not invert).

To list all command line options, use one of these calls: `SS -?` or `SS -help`. More info about the ADMB command line options is available in the ADMB Manual (Chapter 12: Command line options).

To run SS3 without estimation use: `ss -stopph 0`. This will speed up your run by not optimizing. Often `-stopph 0` is used with the `-nohess` option to speed up the run even more. To run SS3 with no estimation in SS3.30.15 and earlier, change the max phase in the starter.ss file to 0 and run the exe with the `-nohess` option.

4.2 Using ss.par for initial values

Typically, SS3 uses initial parameter values as specified in the parameter lines of the control file. However, initial values can be read from the ss.par file instead. To use ss.par, in starter.ss below the names of the data and control files, set the value to 1 rather than 0 on the line with comment:

`# 0=use init values in control file; 1=use ss.par`.

Using the ss.par file is helpful when you want to use different initial values without modifying the control file. An ss.par file is an output file created during each model run and contains the ending parameter values. Using this ss.par file from a previous model run with a similar model (that still has the same parameters) will speed up run time because the initial values are likely closer to the maximum likelihood estimation (MLE) parameter estimates.

4.3 Using wtatage.ss

As stated in the manual, SS3 has the capability to read empirical body weight at age for the population and each fleet, in lieu of generating these weights internally from the growth parameters, weight-at-length, and size-selectivity. Please see the Empirical Weight-at-Age (wtatage.ss) section of the SS3 user manual for more information on using wtatage.ss with a model. There is also a weight at age SS3 example model available if you would like to try using this feature without having to build your own model.

5 Plotting results and basic troubleshooting

5.1 Using r4ss to organize and plot SS3 output

To plot the results of an SS3 model after it has run, use:

```
library(r4ss) # use r4ss package on github
# create list of quantities for the model in the directory mydir
replist <- SS_output(mydir)
SS_plots(replist)# create plots for SS3 run
```

See https://github.com/r4ss/r4ss for r4ss installation instructions.

5.2 What to do when SS3 doesn't run

Here are some basic checks for when SS3 does not run:

- Make sure that all files and directories are named correctly.
- Check that starter.ss references the correct names of the control and data files.
- If SS3 starts to read files and then crashes, check warnings.sso and echoinput.sso. The warnings.sso will reveal potential issues with the model, while echoinput.sso will show how far SS3 was able to run. Work backwards from the bottom of echoinput.sso, looking for where SS3 stopped and if the inputs are being read corectly or not.

For further information on troubleshooting, please refer to the SS3 User Manual "Running Stock Synthesis" subsections, especially "Re-Starting a Run" and "Debugging Tips".

6   Where to get additional help

- The SS3 vlab website resources, including the SS3 user manual and the SSI user guide
- Post questions to the forums
- Email questions to nmfs.stock.synthesis@noaa.gov
- Report bugs or request features using the Stock Synthesis issue tracker
- Fisheries Research special issue (vol. 142) on Stock Synthesis
- Center for the Advancement of Population Assessment Methodology (CAPAM) provides information about good practices in stock assessment

7   Appendix 1: An introduction to the command line

If you have never used the command line before, it will be helpful to learn a few basics. Some potential resources include:

- Appendix A to the Happy Git and GitHub for the useR book, which introduces the command line on Linux, OSX, and Windows systems.
- Introduction to the Command Line by Launch School has even more detail about using the command line specifically on Linux (which should also apply to OSX systems)
- A web search on "Command Line" will reveal plenty of other resources

8   Appendix 2: Putting SS3 in your PATH

Instead of copying the SS3 executable to each model folder, SS3 can be put in your system path, which is a list of folders that your operating system looks in whenever you type the name of a program on the command line. This approach saves on storage space since the SS3 binary (i.e., the SS3 executable or exe) is about 2.2 MB and having it located in each folder can be prohibitive in a large-scale simulation testing study. Even if you are not running a large simulation study, putting SS3 in your path may still be convenient, as you can use the same executable on many models, there is no need to specify a full file path to the executable each time you run a model, and no need to create a batch file that refers to the executable's location.

8.1   For Unix (OS X and Linux)

To check if SS3 is in your path, assuming the binary is named SS: open a Terminal window and type `which SS` and hit enter. If you get nothing returned, then SS3 is not in your path. The easiest way to fix this is to move the SS3 binary to a folder that's already in your path. To find existing path folders type `echo $PATH` in the terminal and hit enter. Now move the SS3 binary to one of these folders. For example, in a Terminal window type:

```
sudo cp ~/Downloads/SS /usr/bin/
```

You will need to use `sudo` and enter your password after to have permission to move a file to a folder like `/usr/bin/`.

Also note that you may need to add executable permissions to the SS binary after downloading it. You can do that by switching to the folder where you placed the binary (`cd /usr/bin/` if you followed the instructions above), and running the command:

```
sudo chmod +x SS
```

Check that SS3 is now executable and in your path:

```
which SS
```

If you followed the instructions above, you will see the following line returned:

```
/usr/bin/SS
```

If you have previously modified your path to add a non-standard location for the SS3 binary, you may need to also tell R about the new path. The path that R sees may not include additional paths that you have added through a configuration file like `.bash_profile`. If needed, you can add to the path that R sees by including a line like this in your `.Rprofile` file. (This is an invisible file in your home directory.)

```
Sys.setenv(PATH=paste(Sys.getenv("PATH"),"/my/folder",sep=":"))
```

8.2   For Windows

To check if SS3 is in your path for Windows: open a DOS prompt (either Command Prompt or Powershell should work) and type `SS -?` and hit enter. If you get a line like `SS is not recognized...` then SS3 is not in your path (assuming the SS3 executable is called SS.exe). To add the SS3 binary file to your path, follow these steps:

1. Find the correct version of the `SS.exe` binary on your computer (or download from the SS3 releases).

2. Move to and note the folder location. E.g. `C:/SS/`

3. Click on the start menu and type `environment`

4. Choose `Edit environment variables for your account` under Control Panel

5. Click on `PATH` if it exists, create it if does not exist

6. Choose `PATH` and click edit

7. In the `Edit User Variable` window add to the **end** of the `Variable value` section a semicolon and the SS3 folder location you recorded earlier. E.g. `;C:/SS`. **Do not overwrite what was previously in the PATH variable**.

8. Restart your computer

9. Go back to the DOS prompt and try typing `SS -?` and hitting return again.