Developing your first Stock Synthesis (SS3) model

SS3 Development Team

Last Updated: September 18, 2025

1 Scope

The developing your first SS3 model guide teaches users how to develop a basic Stock Synthesis model. We assume that these users have had previous population dynamics modeling experience and already understand how to run an existing SS3 model.

If you are a new SS3 user who is not yet comfortable running an SS3 model, we suggest trying to run an example working model using advice in the Getting Started guide before attempting to develop and run your own model as outlined here.

By the end of using this guide, you should be able to:

1. Develop your own SS3 model development workflow based on the tools and methods suggested
2. Find information on the required inputs for different model specifications
3. Understand how to use phases within an SS3 model
4. Debug basic issues with input in Stock Synthesis models and identify when there are more advanced model problems

2 How to create new SS3 models

There are many potential workflows for developing a new SS3 model, but a common technique is to start with an existing model and change it in a piece-wise fashion. The basic technique is:

1. Find an existing working model (perhaps one of the SS3 example models available in the ss3-user-examples repository or perhaps a model more similar in form to the model you plan to create).
2. Edit portions of the model and try running as you go to check that your inputs can be read correctly by SS3 and make logical sense (tip: use options `-stopph 0 -nohess` to reduce run time with each iteration of checking inputs by not estimating anything and not inverting the hessian).

Some commonly used tools for editing the SS3 input files are:

1. **Stock Synthesis Interface (SSI; the SS3 GUI)**. The SSI allows you to read in a model, performs some checks to ensure valid inputs, make modifications to the model, and offers visualizations of inputs. You can also run models from SSI. Note that SSI is not maintained for Stock Synthesis versions after v.3.30.21.
2. **Your favorite text editor**.
3. **The `SS_read*` and `SS_write*` functions in the R package r4ss**. These functions allow you to read in SS3 input files to R, manipulate them from within R, then write them out to a file. The r4ss vignette demonstrates how to use these functions
4. **Stock Assessment Continuum Tool**. Available through github at https://github.com/shcaba/SS-DL-tool.

3 Guidance on model specification

SS3 has a rich set of features. Some required inputs are conditional on other inputs, so it is important to be mindful that changing an option might result in SS3 expecting a different number of values in the input files than it did previously. Most of the time, these conditional inputs are read right after the option.

The SS3 user manual can be used as a guide to help you edit your model. Conditional inputs are noted in the manual. The SSI can also help guide you through changes in model inputs required as you select different SS3 model options.

If you are unsure if you got the setup right (e.g., adding the correct number of parameter lines for a chosen catchability setup), try running the model with `maxphase = 0` in the starter file and ADMB option `-nohess` (or for v.3.30.16 and greater, run the model with command line options `-stopph 0 -nohess`, no need to change the starter file). If the model run completes, you can compare the **control.ss_new** file and the first data set in **data.ss_new** to your SS3 input files to make sure SS3 interpreted the values as intended. If the run exits before completion, you can look at **warning.sso** and **echoinput.sso** for clues as to what was wrong with your setup.

For additional help with model specification, please post your questions in the GitHub discussions or on the vlab forums (for registered SS3 users) or send an email to the SS3 team at NMFS.Stock.Synthesis@noaa.gov.

4 Phases in SS3 and ADMB

Phases are used within ADMB to tell the program in what order the parameters should be estimated. For more details about phases in ADMB, see the ADMB user manual. ADMB phases can be specified for parameters in SS3.

A **negative phase** indicates that the parameter is not estimated, but specified. Note that all negative phases are equivalent; i.e., there is no difference in how parameters with phases -1 and -99 are dealt with. The exceptions are:

- -9999, which is a special value within SS3 (see the profilevalues.ss section of the SS3 user manual).
- Values more negative than -999 for parameters in the tagging section of the control file.

A **positive phase** indicates that the parameter will be estimated. Parameters with a lower positive phase will be estimated before those with a higher positive phase. Which phase should you put an estimated parameter in? While there is variability among analysts and models in how phasing is used, in general, parameters with low positive phases should be for values which define the scale of the population (i.e., *R0*, natural mortality, steepness, selectivity slope). Parameters for fine-tuning fits to the data are often estimated in later phases (i.e., catchability, growth, and recruitment deviation parameters). Many models use positive phases from about 1 to 5, although it is possible to use higher phasing values.

When a model is run without estimating anything, the phases are ignored; because there is no estimation, running without estimation is like including a negative phase for all parameters.

5 Basic debugging: use echoinput.sso and warning.sso

The most useful tools for debugging SS3 issues are the SS3 output files **echoinput.sso** and **warning.sso**. As SS3 runs, it writes back to **echoinput.sso**. When SS3 exits without finishing, looking at the last contents written to **echoinput.sso** and comparing it with your input files can provide clues as to where (and possibly why) the run failed. Note that SS3 reads inputs in order and will read them incorrectly if the wrong number of inputs is provided. Any warnings generated during the run are written to **warning.sso**. The **warnings.sso** file should always be examined at the end of the run, even if the run was successful, as it includes suggestions for improving the model specification and warnings about common model sepecification mistakes.

6 Visualizing results

Visualization is helpful as you iteratively develop your model and for presentation of final model results. The output of an SS3 model can be read into r4ss. The r4ss vignette has more information on how to use the r4ss R package.

7 Proposed workflow for arriving at a "final" model

It's important to remember that arriving at a model takes time and many iterations! You should plan on making small revisions, rerunning the model, and looking at how the model has changed with each small revision. Typically, many intermediate model runs are required before you reach a final model. At a minimum, keep notes inside of the run folder or elsewhere regarding the run specific changes. Using version control could also be helpful for tracking model changes. Below, we provide some suggestions on the steps to take as you iterate toward your "final" model.

7.1 First runs without estimation

First, check that SS3 reads the input files as intended. Use `-stopph 0 -nohess` command line options to run the model without estimation. Then examine **warning.sso**, **echoinput.sso**, and the **.ss_new files** produced to see how SS3 interpreted your input files and if it was as you intended.

After you have verified that SS3 is reading your input correctly, you can also consider adjustment to selectivity, growth, and recruitment parameters.

7.2 Runs estimating some parameters

Try changing max phase to be greater than 0 (perhaps 2 or 3) to estimate some but not all parameters. Do this by using `-stopph x -nohess`, where x should the maximum phase desired (e.g., 2 or 3). `-nohess` is still used to reduce run time. During this stage of model runs, you should try to adjust the model to resolve all major patterns in the residuals. This may include considering and making changes to the model structure.

7.3 Runs estimating all parameters

Once all major patterns in the residuals have been resolved, try estimating all parameters in the model. Change max phase to include all of your model phases (i.e., equal to or higher than the highest phase specified in your model). You'll want to do multiple runs for different model refinement purposes at this time:

1. Do one or more run(s) with the hessian estimated to get variance of the recruitment deviations and then use this vector of variances to adjust the bias adjustment ramp (see the SS3 User Manual for additional details).
2. Do runs with or without estimating the hessian to tune the data weightings.
3. Run the model multiple times and jitter the starting values (option to jitter is selected in `starter.ss`). This runs the model with new starting parameter values to determine if a better model fit is possible using different starting values.

After these runs, your model should be fine-tuned and ready for final runs.

8 Advanced debugging

In this section, we have listed some common problems that you may encounter and some techniques to resolving them within your model.

8.1 If the residuals have substantial patterns

It is often difficult to pinpoint the cause of residuals that have substantial patterns, but two potential reasons this could happen are unrealistic starting values or poor model specification. If you have residuals with substantial pattern, you can try changing the starting values. If a parameter is not moving from the starting value, there may not be enough information to determine its value and it may be necessary to fix the value in the model. If the parameter does move with different starting values, it is possible that the starting value used before was not reasonable and thus changing the starting value to a reasonable value fixes the issue.

8.2 If SS3 runs, but there is a large gradient at the end

Large gradients are usually reported in **warnings.sso** file, so this problem should be clear when examining the file after a model run. If this is the case, try the following

- Examine **Report.sso** for parameters that are on or near bounds, as these may be the parameters that are causing issues. It depends on the value, but you can consider changing your bounds or the parameters, depending on what seems most plausible given your model.
- Examine residuals to identify if there is overall poor lack-of-fit. This may indicate that the ADMB algorithm failed to find the global minimum. Try alternative starting values to attempt to solve this problem. However, it is also possible that there is large disagreement between your model structure and your model file, in which case you may want to reconsider the model specification - perhaps there is an alternative model structure that makes more sense for the population.

8.3 If the hessian does not invert

The hessian information is reported in the **Report.sso** file, so go to this section to confirm if this is an issue. This is a difficult issue to solve, but considering alternative models may be necessary if the hessian will not invert.

9 Advanced ADMB run options (MCMC, adnuts)

SS3 models are typically run using maximum likelihood estimation (MLE), but it is possible to run SS3 models using Markov Chain Monte Carlo (MCMC). MCMC is outside the scope of this document, but below are some resources to check out in case you are interested in pursuing it further:

1. See the ADMB user manual for more information on using MCMC in ADMB and the section on Running MCMC in the SS3 user manual for more specific information on running MCMC in SS3.
2. See the adnuts repository for more information on using the No-U-Turn sampler (NUTS) in ADMB and this article by Monnahan et al. (2019) to better understand the benefits of using NUTS for assessment models.

10 Some questions to consider during model development

There are many elements to consider in an SS3 model. Below are some questions and considerations that experienced SS3 users have used to check that their model makes sense throughout the iterative model development process.

10.1 Does my model model make sense?

- Does the biology make sense? Selectivity?
- Is the model fitting the indices? Length and age data?
- What is the pattern in recruitment?
- Does the prior type match what I thought I was specifying?

10.2 Does my model model fit well?

- The `r4ss::SS_output()` function prints some very useful diagnostics to the R console window, including a list of the estimated parameters, the estimated value, whether the parameter is hitting a bound, the gradient, and the prior type. Use this (or other SS3 output files) to answer:
  - Did the parameter move from the initial value?
  - Is the parameter hitting a bound?
  - Is the parameter gradient low?
- Below the parameter estimates that `r4ss::SS_output()` prints to the console, there is a specific section that highlights the parameters with the highest model gradients. Examining this list might indicate which (if any) parameters may need to be fixed due to lack of information in the data. For example, there may be selectivity parameters included in the list where small changes in the parameter value do not change the model results (or the form of the selectivity).

10.3 Does my data weighting make sense?

The `r4ss::SS_output()` function prints information on tuning to the R console. This includes tuning information for length and age data that can be used for data weighting when using McAllister Ianelli weighting (i.e., weighting by the harmonic mean). The `r4ss::SS_plots()` function creates mean length and ages figures with captions showing the recommended Francis weights for each data source. Even if you are using an alternative data weighting approach (i.e., Francis, Dirichlet, or other), it is always good practice to explore an alternative approach and see how data weighting changes the model.

10.4 Did I fix variation in recruitment correctly?

Check the sigmaR (i.e., recruitment devs standard deviation) information. The sigmaR parameter is typically fixed within the model, so if it is not fixed within your model, you should consider whether or not this makes sense for the population and given the quality and quantity of data.

For a fixed value of sigmaR, a section of the output will provide diagnostics to determine if the fixed sigmaR value is capturing the estimated variations in recruitment. The `r4ss::SS_output()` function will print a recommended value based on the variation in the estimated recruitment deviations that you may want to consider using.

11 Where to get additional help

- Post questions in the GitHub discussions or to the forums or send an emails to the team at NMFS.Stock.Synthesis@noaa.gov for assistance.
- Carvalho et al. 2021 contains guidance on developing stock assessments.