

## How to evaluate an SSL block

1. Check for any IgnoreBlock() or RequireBlock() commands. If you find an IgnoreBlock() any of whose arguments are "True", or a RequireBlock() none of whose arguments are "True", discard the whole SSL block and move on to the next block.
2. Take the block's list of triggers and go through it in order. Remove each Target, TargetBlock and ConditionalTargetBlock command, and put them aside. For each argument of each TriggerBlock command, check the libraries for a trigger list with that name as argument. If you don't find one, FAIL the whole SSL run. If you do, insert all the triggers in the list at that point in the block's list of triggers. At the end of this process, you'll have a list of triggers (call this the Specific Trigger List) as well as a list of Target (et al) commands. *If there are no Target, TargetBlock or ConditionalTargetBlock commands, insert Target(NODEFINEDTARGET).*
3. Take the block's list of actions. Set aside the Action() and ActionCondition() commands. If you find Combine(), make a note that this is a Combine() run. If you find OnContinue(), extract the BAF actions from its argument and make them into a list (call it the Continue List). The residue after doing this is a list of BAF actions; call this the Specific Action List.<sup>1</sup>
4. For the moment, *assume that this isn't a Combine() run.* (It's easier to explain Combine() syntax later). Go through the Action and ActionCondition blocks in order. If you've got an ActionCondition block, extract the conditions and temporarily add them to the trigger list (i.e., take them away after evaluating this action). Then look for the action itself in the list of already defined actions. If you can't find it (unless it's the internally-defined action "Literal"), then FAIL the whole SSL run.
5. Assuming you can find the action, it should give you something that has the skeletal form of a BAF block.
  - a. In that skeletal form, substitute each scsargument variable with the value it's been defined to have in the Action (or ActionCondition) block. Replace any backslashes in the argument with commas. If scsprob1 or scsprob2 has been defined, substitute them too; otherwise, set scsprob1 to 100 and scsprob2 to 0 and substitute them.
  - b. If there are any ordinary BAF actions defined by the original SSL block, append them to the end of each action block in the skeletal form.<sup>2</sup>
  - c. If scsprob2>0, add an additional action block, RESPONSE #scsprob2 Continue().
  - d. If there's a Continue List (cf. 3), add it to this block before the Continue() command.
  - e. Finally, take the Specific Trigger List and append it to the end of the triggers in the skeletal block.

---

<sup>1</sup> I've no idea what SSL does with multiple OnContinue() or Combine() blocks. Probably fail.

<sup>2</sup> Actually, SSL doesn't quite do this: it just appends them to the *last* such block. That's unintentional, though.

6. After step 5, you should have something that looks exactly like a BAF block, *except* that the weird “scstarget” object keeps cropping up. (If not, something’s gone wrong). Now go through the Target, TargetBlock and ConditionalTargetBlock instructions.
  - a. For each Target() instruction, and each argument of that instruction, substitute the target for “scstarget” in our proto-block, and then output it.
  - b. For each argument of each TargetBlock() instruction, look for that argument in the library files. If you can’t find it, FAIL the whole SSL run. If you can, it should give you a list of objects. Treat each as the argument of a Target() command, and generate one block per object.
  - c. For each ConditionalTargetBlock() instruction, temporarily append the condition (a BAF trigger list) to the trigger list in our proto-block (i.e. take those triggers away again once the ConditionalTargetBlock has been evaluated). Then treat it as a TargetBlock()
  
7. Now let’s talk about Combine(). The idea of a Combine() command is that it lets lots of Actions use the same targeting. Normally, if there are N targets and M actions in an SSL block, SSL generates NxM BAF blocks. With Combine(), it generates N+M BAF blocks.
  - a. To generate the N trigger blocks, go through the M action definitions and collect only those triggers in the actions which are common to all M actions. Call this the Common Trigger List. Then assemble this block:
 

```

IF
    [common trigger list]
    [specific trigger list]
    False()
THEN
    RESPONSE #100
    NoAction()
END
          
```
  - b. Take the resultant block from (a) and apply step 6 to it. *Then reverse the order of the resultant blocks, so that the highest-priority Target appears last, not first.*
  - c. Go through each of the actions and assemble blocks for them as if the only Target command was Target(LastSeenBy(Myself)). Add a See(LastSeenBy(Myself)) to the end of the trigger list of each of the resultant blocks.

Note that using ActionCondition inside a Combine() will probably give weird and undesirable behaviour.