

ToDo

1. for all classes that are native to both AS and JS (like Error, Date etc.), check if the properties and methods on both sides match up
 - a. ?
 - b. EdB: there are classes (like Date) that exist natively in both AS and JS, I wrote this ToDo to remind me to check if not only the name, but also the implementation matches.
2. add support for multiple arguments in the 'trace' utility function
 - a. ?
 - b. EdB: the native trace function takes multiple arguments which it concatenates together; the workaround function I propose on the pages below doesn't, needs fixin'
3. can we create a "Vector" class simply by 'extending' the JS Array and using annotation to get the 'type safety'## ?
 - a. EdB: as a Vector is a typed Array, I was thinking about using the JS Array object and subclassing it into 'Vector', that's all
4. is the regular expression syntax the same between the two (AS and JS) implementations?## ?
 - a. EdB: various implementation of RegExp in different languages tend to have slightly implementations, we need to know if that's also the case between AS and JS
5. Mike, can we make the distinction between XML and 'regular' operators etc.?## Yes, we can make this distinction## EdB: ok, at some point we need to figure out the whole "XML" thing, do we just throw exceptions each time we find an XML related item, or what?
6. Research the need to (Closure) compile the JS with the '--language_in=ECMAScript5' argument (defineProperty)
7. ...
8. profit!!!

Code Refactoring

- move all emitting code into emitter from ASBlockWalker, I have no idea how this will affect performance but, it's not my main concern right now, the duty as it stands for the walker is to traverse and call public API on the emitter.
- Figure out how a configuration can be passed into the walker and emitter decoupled.
 - The problem here is that the main Configuration class needs to be fully initialized, when we are writing unit tests, we are not actually running the compiler, we are testing units.
- I have to look at the visitor switch since I wrote that pattern a while back and I think enums are treated different now. I have done investigation about instanceof and Java and it seems like the consensus is it is just as fast as a switch in the new versions of Java. I think we should try to write a performance test using a couple different algorithms IE instanceof, switch on node id before we change it though.
- Finish the ActionScript API XML, and whatever is left in the main code emitter.
- Make internal classes emit.
- Make package functions emit.
- Get asdoc handler saving the asdocs of the members so we can turn that on or off if we want comments on the original source code.
- Figure out if we can write a custom tokenizer that could get single line comments if you need them, do you really need single line comments?
 - ?
 - EdB: don't think it's top (or even moderately high) priority, but it sure would be nice if a developer could 'carry over' a comment to JS

Questions from TODOs

- why aren't comments preserved?
 - The AS3Parser does not save tokens, thus as it stands inline comments are discarded
 - ASDoc comments need a special handler that has to be implemented by the client, I have done this already with the new ASDoc program
- handle interfaces and accessors first
 - What do you mean?
 - EdB: this means that before I could fix the test with that comment, I need to finish the code that handles interfaces and accessors
- why do we need to put use an 'actual' component
 - ?
 - EdB: it looks like that when I put a 'fake' class name after the 'extends' keyword (like 'myComponent'), the JSDoc notation '@extends' isn't added; if I put a real component name there (e.g. spark.components.Button), it shows up fine in the JS. I my limited understanding of unit testing, I think this kind of requirement (having actual framework present) is 'a bad thing', correct? Not sure if and how to work around this...
- handle workaround for "for-each" loop
 - ?
 - EdB: JS doesn't have a 'for-each' loop, so I need to put a workaround in the compiler (like we did for default argument values)
- how to handle the escaping of backslashes
 - this takes a bit of wizardry, I will work on it IE RegExp tests