

## Notes - 10/19 - SENT

Friday, October 19, 2012 8:44 AM

[In 41/2731 at 10:00 today.]

### Agenda:

- Implementation update: VS debugging using source maps demo
- Overview of closure type system from Steve
  - Decision: Add minification to feature list
  - Decision: Encourage community projects to do migration from other formats
- Design backlog
  - Fundules, clodules and lightweight mixins
    - Properties on functions
      - function foo() {}
      - foo.data = "hello";
    - Nested classes
      - class Geometry {}
      - class Geometry.Point extends Geometry {}
    - Namespace interfaces
      - interface jQuery.AjaxSettings {}
    - Concepts in TS:
      - Interfaces
        - T
      - Modules
        - N + T + P
      - Classes
        - T + P
      - Functions
        - P
      - Variables
        - P
    - Names can be:
      - Types
        - Things that can go after colon
      - Namespaces:
        - Things that can contain namespaces or types
        - N.N.N.T
      - Properties
        - Things that are in the value scope referring to things at runtime
        - P.P.P.P
    - Examples:
 

```
interface jQuery {
}
module jQuery {
  interface XHR { ... }
}
var jQuery: ...;
```
    - Proposal:
      - Module, classes, function can all contribute to the same name
      - If only 'module' definitions contribute to a name, it can be spread across files
      - If any class and/or function definition contributes to a name, all contributions must be in the same file
      - Allow dotted named declarations
 

```
function x.y(...)
```
      - Consider: When we know a name exists, don't generate the potential creation code
      - Bring back anonymous modules??
 

```
IIFE
module {

  // 1000 lines of code

  class Bar implements SignalR { }
  export interface SignalR {}h
  export interface $ {
    signalR: SignalR;
  }
  $.signalR = new Bar();

}
```
    - Brands
      - Provide:
        - Know it was created by your constructor
          - This isn't that important
        - Captures knowledge of privates
          - This is partly handled for free given privates as members of the type
          - But need to make sure privates have unique names (GUID attached)
    - Summary?
      - Namespacing types
      - Extending functions/classes with modules
      - Getting rid of brands

- Open ended classes
- Anonymous modules

- Workaround for Monaco to force a file to be treated as external module (#95)
- ... for arguments
- “boolean”? (#135)
- Better typing for Function.prototype.bind (#133)
- Static initialization (#74)
- Generics, map out requirements (#185)
- Thinking on how to approach async via generators (#38)
- String interpolation
- SkyDrive feedback – implications for future thinking?
- Others?