

ECMAScript Standards Meeting  
Held 30<sup>th</sup> October 2001,  
Microsoft, Redmond, WA

### **Attendees**

Waldemar Horwart, Netscape  
Jeff Dyer, Mountain View Compiler Company  
Eric Lippert, Microsoft  
Herman Venter, Microsoft  
Peter Torr, Microsoft

### **Agenda**

Errors in the switch algorithm  
General document discussion (no minutes)  
ECMAScript MIME type  
Operator overloading  
Characters  
Numerics

### **Errors in the switch Algorithm**

The algorithm for the switch statement in Edition 3 is badly broken. Waldemar has updated the algorithm and has tested that it conforms to the behaviour of both Netscape's JavaScript and Microsoft's JScript implementations. The amended version will appear in Edition 4 and will also be added to the Edition 3 Errata document.

### **ECMAScript MIME Type**

Björn Höhrmann (derhoermi@gmx.net) has created an Internet Draft describing a MIME type for ECMAScript and related languages; the draft can be found at <http://www.ietf.org/internet-drafts/draft-hoehrmann-script-types-00.txt>. We briefly discussed this and decided that two MIME types would be appropriate:

- application/ecmascript for existing content based on Editions 3 and earlier
- application/ecmascript4 for new content based on Edition 4

The specification should also point back to the official ECMA site for the specification of the language. I have contacted Björn about our decision.

### **Operator Overloading**

We discussed this topic some more, finally deciding to model Edition 4's overloading on that of C#'s. Operators will be static members of the class, and at least one of the parameters must be of the defining type. There are no explicit pre- or post- fix operators; instead, the operators must be designed to perform correctly under both circumstances, and it is up to the caller to preserve the correct semantics. The overloading rules for ECMAScript are different to those from C#, and the committee will need to decide on default behaviour for Object (as for Edition 3's ToString, etc.)

### **Characters**

Both Microsoft and Netscape allow indexing into strings with the [] operator, but Netscape's implementation currently returns a single-character string, whilst Microsoft's implementation returns a character. Microsoft also considers a character to be strictly equal to a string containing only that character, as determined by the === operator. This is primarily a convenience for users of the switch statement, as there is no syntax for a character literal in the language. Netscape will consider these two differences.

### **Numerics**

(Note this has been superseded by more recent decisions)

We discussed having an infinite-precision numerical type for storing intermediate values during computation, instead of using double as in Edition 3. This would not be a type exposed at the language level, but would be used internally during calculations; nevertheless, it would make a difference to user code as calculations involving large integers (or large intermediate results) would be more accurate. Microsoft's implementation currently overflows to double.

Netscape's current behaviour for the implicit conversion of an integer type to a smaller integer type is to truncate. Microsoft's implementation generates compile-time warnings and run-time errors if overflow is detected, but performs truncation if an explicit cast is given. In the future, we may consider a feature similar to checked / unchecked in C#, where a programmer can specify whether to truncate or throw on overflow. The default should be to throw, as users who do not care about such issues will probably not be using types in the first place.