ECMA International

ECMA/TC39-TG1/2002/10

Standardizing Information and Communication Systems

ECMAScript Meeting 7th August 2002

Microsoft Corporation.

Attendees

Waldemar Horwat (Netscape)
Steve Adamski (Netscape)
John Schneider (BEA)
Jeff Dyer (Compiler Company)
Rok Yu (Microsoft)
Peter Torr (Microsoft)

Agenda

Decimal
Reflection
Hawaii Meeting
Schedule
Document process
Document changes

Decimal

Briefly discussed history of Decimal in the ECMA committee and how it may apply to the XML work. Decided it was not enough of a motivating factor to include it in the standard.

Reflection

Basic goal is to make common use of **for...in** for prototype-based objects. Microsoft chose to not make class members "indexible" for various reasons. Expando properties and old-style objects are enumerable. We kept **for...in** only for collections / arrays / old-style objects and did not overload it for reflection purposes.

Problems: What happens with overloaded methods, properties, etc? Non-public things?

Waldemar wants:

```
a["b"] === a.public::b
```

Which seems appropriate. If there is already a **private::b** then the **public::b** would not clash with it, but if there was a **public::b** it would be accessed. Only publics are enumerated, not internal or protected things. It will look over members before the expando table. This will work for all classes except those that override square brackets.

JScript currently takes dot to mean "guarantee that this member exists", whereas square brackets means "look up at runtime". The standard relaxes this rule if the class is expando so that dot doesn't guarantee the member exists (except if it is of a known type which is marked as final). JScript will probably have warnings / errors / slow mode / compatibility mode / etc to match this behaviour.

What is enumerable? On instances, you get all public expandos, fields, properties, and methods. Members of the base class are enumerated. Don't know about events, operators, conversions, attributes, etc. On types, public static expandos, fields, properties, and methods are enumerated. The type **Type** itself has no members (either instance or static) as defined in the standard. Tentatively: A type may contain static expandos (eg, String) but there is no way for a user to define such things.

Issue: Currently Jscript does not allow access to base-class statics from derived classes.

There is no way (in the standard) to enumerate the properties, fields, or methods of something that is a collection or array. Put this off to Ed 5. Expandos work as expected though.

Reaffirmed previous decisions to postpone operator overloading and user-defined coercions to v 5. Cut "enumerable" since that is now the default, and postpone "non-enumerable" to vNext.

Next meeting - Hawaii

Have the ECMASript meeting on Wednesday 2nd in the morning, with XML on Wednesday afternoon / Thursday morning. The general meeting is Thursday afternoon.

Schedule

Edition 4 still looks good for December 2003 GA, as does the XML work (standard must be done by September to be accepted in December). The XML work should be done before September, but not in time for the June GA (March). Unclear on how the XML stuff will integrate (addendum, 4.5, integrated, etc). Do not want to risk Edition 4 schedule with XML integration work. John is happy to do the XML work at a pace comfortable with the group. It is important to get it out, but there are no pressing deadlines.

Document process

Currently coming to "functional agreement" -- agreeing on what features should be in the standard, and how they should work at a relatively high level. Need to move to "content complete". Rok believes we need to build a table of contents for the document and mark the completeness of each section and who has reviewed it to aid in tracking. He will look into getting this done and find somewhere to publish / share it.

Document changes

Chapters 10 and 12-16 have changed a lot to remove units and operator overriding and to add long, ulong and char types. Added object literal syntax to the document. Need to add characters to section 9.1 (defined in 5.8). Chapter 7 (Lexical grammar) has not been updated for units being removed yet.

Waldemar wants raw numeric literals to be parsed as doubles. Literals that are suffixed with "l" or "ul" denote long or ulong literals, respectively.

Microsoft's assumption is that users don't really want / need longs, but some parts of the runtime use them (including large enumerations) so we added support. We don't in general believe people do arithmetic on longs that requires the precision to be maintained. Netscape does not want to silently generate incorrect results due to loss of precision.

Rok will try and summarise Microsoft's complex rules for JScript and see where we can simplify / compromise.

Waldemar does not want comparisons between numbers to do conversions or coercions. Currently Microsoft does this (eg, a long and a double will compare equal if they are "approximately" the same -- we convert the long to a double, which loses precision). Microsoft will look at changing it.

ECMAScript XML Subgroup Meeting, 8th August 2002

Microsoft Corporation.

Attendees

Waldemar Horwat (Netscape)
Steve Adamski (Netscape)
Michael Shenfield (?)
John Schneider (BEA)
Jeff Dyer (Compiler Company)
Rok Yu (Microsoft)
Peter Torr (Microsoft)

Agenda

The meeting was conducted by John, based off his PowerPoint presentation; please see the PPT for more information.

Terms of Reference

What edition of ECMAScript should the XML standard target? Although we would like new implementations to be on Edition 4, we also do not want to hold up the XML standard indefinitely if Edition 4 is delayed, since mobile developers (compact profile) would like to use XML with Edition 3. Waldemar would like to target Edition 4 and retro fit for Edition 3 as necessary, rather than the other way around. Rok would like to do the general design first, then determine what features can be supported on each edition of ECMAScript and work on building the standard as appropriate.

Since the Edition 3 work is time critical, we decided to make the XML standard a separate specification (not part of ECMAScript 4) with the intention of rolling it into a future version of the standard (Edition 5?).

Naming

The all-important naming exercise took quite some time and was highly amusing. In the end we settled on **ECMAScript for XML (E4X)** as the name of the standard.

Task Breakdown

One important aspect of ECMAScript has been that we always worked with a prototype / reference implementation when building the standard. It would be highly desirable to have such an implementation for E4X as well. Although there will not be a formal 'reference' implementation, individual companies will be building prototypes to gain practical experience of implementation and to provide input into the standard.

It was decided that we would disclose information about E4X via the web (see next topic)

Timeline

Originally John had planned a single staggered approach to finishing the standard. After some discussion we agreed that having a two-phase process might be better; an initial high-level validation pass to get a complete picture of the standard, followed by a second pass to finish out the details. This would enable the standard to be defined with a holistic understanding of the entire problem area. After the first phase is complete (early 2003?) we will consider opening the standard up for public review, potentially with some prototype implementation(s) available for experimentation.

Next Meetings

Hawaii: October 2-3
Mountain View, CA: November 21-22
Redmond: Jan 15-16

Use Cases

Rok, John and Jeff will think about use cases for the standard before the next meeting. Michael will also try to investigate use cases for mobile devices. Rok will then build a table mapping use cases to features.