

ECMA Working Group Meeting
April 21, 2005

Hosted by Google in Mountain View

Waldemar Horwat
Bill Schulze
Jeff Dyer
Francis Cheng
Gary Grossman
Edwin Smith
Brendan Eich
Rok Yu

Status of ISO Draft

- The ISO draft is mostly done, mostly thanks to Francis. Jeff filled in some content items that Francis wasn't around for.
- Rok has one outstanding action item.
- Brendan says that 10.2.1 has a spec bug. It may need to be errata, since it wasn't in the original comments.

Two parts to this meeting:

- Scrub technical changes to E4X specification.
- Discuss the ECMAScript 4 specification.

E4X Technical Changes

- 9.1.1.1: Rok is working on this one. Fix the MapInfoItem mapping. Touches the spec in several places; should not be rushed.
- 9.1.1.2 Step 1, 2, 3, 4: Accepted
- 9.1.1.2 Step 6: Accepted
- 9.1.1.2 Step 9: Editorial change, not technical
- 9.1.1.3, 9.2.1.3 and elsewhere: Make the use of conditional statements consistent. This change is being rejected, because the risk/reward is not there. When the comment was originally made, it was thought to be more widespread in the document, but it's not.
- 9.1.1.4, 9.1.1.13 are editorial changes, not technical
- 9.2.1.2: "To match insertChildAfter, ..., we should silently do nothing in this case." Accept. Brendan reviewed the change.
2.c.ii. 1. If r.[[Class]] != "element" 1. Return

- 9.2.1.2: Editorial changes skipped
- 9.2.1.2: Step 2(c)(vii)(3) accepted
- 9.2.1.2: Step 2(f)(iv, vi).
 - Replace (iv) and (vi) with these steps:
 - iv. For $j = x.[[Length]] - 1$ downto $i + 1$, rename property j of x to `ToString(j+c.[[Length]]-1)`
 - vi. Let $x.[[Length]] = x.[[Length]] + c.[[Length]] - 1$
- 9.2.1.2: Aside from the original comment, we missed a special case when $c.[[Length]] = 0$. In that case, the items need to be shifted left, not shifted right, because an item is being deleted, not inserted. There should be something like:
 - If $c.[[Length]] = 0$,
 - For $j = i + 1$ to $x.[[Length]] - 1$, rename property j of x to `ToString(j-1)`
 - Else
 - For $j = x.[[Length]] - 1$ downto $i + 1$, rename property j of x to `ToString(j+c.[[Length]]-1)`
- 9.2.1.2: Brendan asked what the purpose of the loop in Step 2(f)(iii) is. In his implementation, he is just asserting pointer identity instead of actually copying the elements. He believes this was present due to part of XMLBeans.
 - Brendan is worried that this might make it possible for the ISO spec to make cycles.
 - Jeff proposed making this an errata and addressing it in Edition 4.
 - **Action item for Brendan**
 - Brendan gave this example of a possible cycle:
 - `x = <x><y>hello</y></x>;`
 - `y = x.y;`
 - `y.appendChild(x);`
 - `x.toXMLString(); // iloop or recursive divergence, depending on implementation`
 - 9.2.1.2 2(g) calls `[[Replace]]` on the parent of the list's target, and that's enough to make a cycle.
- 9.2.1.2: Step 2(g)(iii). Fairly uncontroversial.
 - Rok thinks we don't want to be calling `ToXML` there, since it will do a parsed conversion from string and we really just want to escape the value / create a text node.
 - **Action Item for Francis:** Work Item: Use the text in 9.1.1.12 7(b) as the model for 9.2.1.2 Step 2(g)(iii)1.
 - Rok: Only difference is you wouldn't set the parent to null.
- 9.2.1.2 Step 2(e)(i,ii) 9.2.1.2. Step 7(e)(i). 9.2.1.3 Step 2(b)(ii)(1)(a). Wrap the arguments in `ToAttributeName`.
- 10.2.1: Step 3 is premature. Change accepted.
- 10.2.1: Step 11: Leave it. Change rejected.

- 10.2.1: Step 12: Change accepted.
- 10.2.1: Step 17(e-f): Change accepted
- 10.2.1: Step 21(a) is editorial change
- 10.2.1: Step 24(a): Change accepted

At this point, with two dozen or more changes to review, we decided to hit only the items in certain need of discussion so that we can move on to ES4 discussion.

- **Action item for everyone to review offline**
 - 10.2.1.1 step 2(b)
 - 10.2.6.1 step 1
 - ... and all remaining ones except the two that will be discussed below ...
- Jeff will set up a conference call in a week to close this out.
- 11.1.4: Grammar. Intent is to tighten grammar so it catches syntax errors in literals, instead of waiting until runtime. Less important for it to be precise than it is for it not to throw an error for valid XML. Instead of just scanning for a significant character, now it has some structure. New token for lexical name.
 - Rok: Do we need to call out actual characters for XMLNameStart?
 - Jeff: Can get rid of attribute conflict by getting rid of syntactic production. Addresses fact that there are two XMLAttributeValues, one lexical and one syntactic.
 - **Action Item for Francis:** Call out the names of the characters.
 - **Action Item for Francis:** XMLWhitespaceOpt needs to be inserted around the '=' sign in the XMLAttribute. XMLWhitespaceOpt needs to be inserted at the end of tags.
 - Rok: I don't think you guys added the productions for all the new things you added? For instance, XMLAttributeValue?
 - **Action item for Francis:** Review all XML productions, make sure they were all added.
- 12.1 Default XML Namespace. All global default xml namespaces are different bindings. You'll inherit the one that would be enforced lexically.
 - default xml namespace = new Namespace("foo")
 - function f() {
 - ... <x/>
 - default xml namespace = new Namespace("bar")

- ... <y/>
- }
- default xml namespace = "baz"
- f()
- Rok: This is only possible if you change the compilation model and object model somehow.
- Waldemar noted symmetry with the "with" statement.
- Jeff: High level goal was to make it behave more like vars.
- **Action item for Jeff:** Jeff proposes that we go back to the drawing board on this one. He will research it.

Schedule conference call to go through updates. Thursday, April 28, 1pm.

ECMAScript 4

Francis and Jeff will get the base document together.

Jeff is pressed for time, but with Francis's help, should be able to make progress.

Rex Jaescke recommended we use the C++ CLI standard as the base. The template and style sheet was iterated over a couple of times, and they think they've got it figured out. Basically the same as the C# standard.

Jeff brings up his ES4 checklist.

Rok is wondering why we're basing off the C++ CLI standard.

Jeff thinks that it makes sense to use a more declarative way of expressing the language.

Rok: This is a lot of work. The question comes up in coding as well, when do you rewrite your code base? Is Edition 4 a big enough change that we need to rewrite the whole spec? I'm worried that we're not going to capture the behaviors. When we review it, we're going to be spending a lot of time making sure we capture Edition 3 behaviors.

Makes sense to prototype it.

Working group needs to have data model, algorithmic detail.

Standard document may have more or less detail.

It needs to be a hybrid; it needs to be a data model that gives the vocabulary.

Action item for Francis: Prototype the basic building blocks of the new spec.

Discussion of what the various features in ES4 are, and assigning owners to come up with proposals for those features.
See Jeff's spreadsheet for assignments and prioritization.
Certain items were marked with "1" priority and have a deliverable for our next meeting.

E4X default xml namespace discussion resumed

Goal is to make default xml namespace behave just like var.
Rok and Brendan are fine with changing it for ISO.