

# *Position on Web Applications and Compound Documents*

**Jon Ferraiolo**  
**Adobe Systems, Inc.**

---

## *Summary Position*

---

- Any new activity should proceed systematically and start with identification of prior related work, identification of use cases, and a statement of requirements.
- Existing W3C specifications already contain most of the protocol definitions that are necessary to address industry requirements in both areas, but there are gaps. Instead of launching major new language definition efforts around either topic, we recommend that the W3C first look at existing W3C specifications, in-process W3C work, and related best-in-class technology efforts that have proven successful in industry. Attempt to codify and integrate what exists today, accept precedent whenever possible, and do not attempt technology invention via standards committee.
- Rich-client web applications often require a combination of multiple W3C languages, particularly XHTML and SVG.
- A key focus must be the mobile industry and emerging alternate devices. The mobile industry, particularly the low end, is still young and market dynamics promote standards adoption. Within the mobile industry, there is strong demand for both web applications and compound documents. The first W3C deliverables in these areas must address the requirements of low-end cell phones and other emerging devices.
- For web applications, the W3C should leverage its existing standards (e.g., DOM, XForms, XHTML, SVG) and all of the applications-related features that are in SVG 1.2, particularly RCC/XBL, the features on the “Window” interface, and the SVG Tiny uDOM that has been developed in conjunction with J2ME and JCP/JSR226.
- For compound documents, the W3C should take a low-ambition approach and codify the rules for existing loose-coupling approaches to compound documents (e.g., <html:object>, <svg:foreignObject>, <fo:foreignObject>). The W3C should not attempt to build new content standards or augment existing content standards in order to work around the lack of a packaging standard.



## *Background and Experience*

---

Adobe has institutional expertise and experience in the design and implementation of document formats for desktop applications (i.e., FrameMaker, InDesign), server tools, and widely deployed clients. We have adopted XML throughout our product line. We develop tools, servers and clients to support both web applications and compound documents. PDF, particularly with its Intelligent Document features, is a web application format and a compound document format. Table 1, below, describes some of Adobe's products and technologies as they relate to web applications and compound documents.

<b>Product or Technology</b>	<b>Description</b>
Adobe Acrobat	Supports web service invocation, export of XML from PDF content, XMP metadata.
Adobe Designer	Recently announced beta program. For creating XML-centric forms applications and intelligent documents targeting PDF or HTML delivery. When using PDF, PDF becomes a compound document format and a web applications format.
Adobe Document Server	Document generation services. Formats a combination of XSL-FO and SVG into PDF. Provides web service interface.
Adobe Forms Server	Forms automation services. Supports XML form data. Provides web service interface.
Adobe FrameMaker	A WYSIWYG XML authoring and publishing solution. Generates templates that support both XHTML and SVG and which feed into Adobe Document Server.
Adobe GoLive	Web application designer. Includes XML editing support. Supports XHTML with embedded SVG. Strong mobile feature set, including MMS.
Adobe Graphics Server	Graphics creation and manipulation services. Renders SVG as raster or PDF.
Adobe Illustrator	Vector-based editing product. Supports SVG import and export. Generates SVG with embedded XHTML-like text flows that feed into Adobe's server products.
Adobe InDesign	Desktop publishing product. Supports import and export XML content.
Adobe PDF	Open specification portable document format which is also (via subsets) an approved official standard. Supports XML form data, XMP metadata, and XDP packaging.
Adobe SVG Viewer	Free SVG renderer for Windows and Macintosh browsers. The version 6 Technology Preview release supports embedded XHTML and RCC support (predecessor for XBL).
XMP	XML- and RDF-based metadata technology that allows metadata to be embedded within various file formats. Freely available specification and SDK.

### **Partial list of Adobe's XML-enabled products and technologies**



---

## *Start with use cases and requirements*

Any W3C activity in these areas should proceed in a systematic manner and start with identification of use cases, identification and collection of prior work, and a statement of requirements. Among the prior work which should be studied, the W3C should review the findings of the ui-tech task force from a couple of years ago. Requirements should take into account issues such as accessibility, multi-modal, and different types of devices, particularly mobile devices, but also take servers into account, where XSL-FO+SVG plays a key role as a common example for compound documents.

---

## *The W3C should focus on codifying and integrating what exists today*

Instead of launching major new language definition efforts around either topic, we recommend that the W3C first look at existing W3C specifications, in-process W3C work, and related best-in-class technology efforts that have proven successful in industry. Attempt to codify and integrate what exists today, accept precedent whenever possible, and do not attempt technology invention via standards committees. As a general rule, avoid inventing new markup languages; instead build on specifications and features that already exist. In fact, often it might turn out that it makes sense to “shrink” existing specifications by consolidating common features of different markup languages into shared namespaces, versus defining new markup.

---

## *Rich-client web applications require both XHTML and SVG*

Adobe has considerable industry experience from its PDF, ePaper, Intelligent Document and SVG initiatives, and we see strong industry demand for what analysts call “rich client” solutions. To a large extent, Adobe’s Intelligent Document Platform is a major strategy push into the rich client arena leveraging PDF, which we have done in response to strong industry demand. A large number of developers are also leveraging SVG for rich web applications. Within the SVG developer community, the largest set of requested enhancements to the SVG language are those that support web applications, where SVG is used as the graphically rich front-end to data-driven applications. Among the most requested SVG enhancements are UI widgets, adjustable layout, and text flows, all of which are found in XHTML.

Usually, the preferred solution for web applications involves a combination of technologies. For applications with reflowable content, XHTML and CSS are often the appropriate standards-based technologies. For applications involving vector graphics or rich user interfaces not possible with XHTML and CSS, SVG is often the appropriate standards-based technology. Very often, the solution requires a combination of both; in other words, the total solution requires some notion of a compound document.

---

## *Strong focus on mobile requirements*

A key focus must be the mobile industry and emerging alternate devices. The mobile industry, particularly the low end, is still young and market dynamics promote standards adoption. There is strong demand in the mobile world for both web applications and compound documents, and a good set of W3C mobile standards will promote new mobile solutions. The first W3C deliverables in these areas must address the requirements of low-end cell phones and other emerging devices.

The most important focus should be integrating XHTML-Basic, SVG-Tiny 1.2 (which includes major subset of SMIL-Basic), CSS-Mobile, a mobile DOM (building on the SVG Tiny 1.2 DOM that is being developed in conjunction with the



JCP/JSR226). Of course, any solution needs to include XHTML. The SVG Tiny module (with all of its SMIL features) is also critical because of the importance of multimedia, infotainment, and location-based services on mobile devices. To meet the requirements for rich user interfaces and mobile web applications, some subset of XBL and XForms probably also will be needed.

## *Web applications*

---

For web applications, the W3C should leverage its existing standards (e.g., DOM, XForms, XHTML, SVG) and all of the applications-related features that are in SVG 1.2, particularly RCC/XBL, the features on the “Window” interface, and the SVG Tiny uDOM that has been developed in conjunction with J2ME and JCP/JSR226.

XBL (fka RCC) provides the low-level framework for custom rich-client user interfaces and provides the mechanism for mapping XForms controls into SVG presentation. All modern application systems include an extensibility UI component framework. An early version of XBL already ships with Mozilla. Microsoft’s Longhorn OS includes the VisualTree facility, which is very similar to XBL. Something like XBL is a necessity for web applications.

Modern web applications need to communicate with servers and other clients. The Window interface in SVG 1.2 is the appropriate low-level starting point for delivering this critical technology. SOAP and Web Services facilities can build on top of this technology.

Mobile web applications have severe memory constraints and thus require a highly subsetted DOM. Strong coordination activities between W3C and the JCP with JSR226, with strong vendor participation among leading mobile implementers, has produced such a subsetted DOM focused for SVG Tiny. Because so many mobile devices ship with Java, the Java binding to this subsetted DOM opens up many opportunities for content creators to leverage W3C open standards in conjunction with Java open standards.

## *Compound documents*

---

For compound documents, the W3C should take a low-ambition approach initially and just codify the rules for utilizing existing loose-coupling approaches to compound documents (e.g., `<html:object>`, `<svg:foreignObject>`, `<fo:foreignObject>`). The W3C should only invent new mechanisms if so forced. Instead of new mechanisms, find the ambiguities and undefined aspects of existing specifications, and fill the specification gaps.

The best long-term solution for compound documents should include a packaging standard as part of the solution set. Many integration issues would be solved if there were a packaging standard which allowed embedded components within the package consisting of XHTML, SVG, CSS, images, scripting and metadata. In particular, the W3C should not attempt to build new content standards or augment existing content standards in order to work around the lack of a packaging standard.