Overview of ECMAScript Components
(TC-39 Study Group item)

Richard Wagner
NetObjects, Inc. / IBM
rw@netobjects.com
Software Components

- Component architecture has been a key software development trend of the 1990s
- Radically change the way software is developed, deployed, and utilized

VBX  ActiveX  Java Bean  COM  CORBA
Benefits of Components

• Simplifies
  – Component is a “black box”, hiding complexity from the component user.

• Adds Power
  – Code modules can “interact” with each other.

• Increases Efficiency
  – Facilitates code reuse.
  – Eases burden of code management.
Components & Scripting

• However, script developers have largely been unable to take advantage these benefits.

• The Results:
  – Higher Learning Curve
  – “Spaghetti Code”
  – “Copy & Paste” Coding Practices
In lieu of a component standard, scripters utilize JavaScript include (.js) files.

**Advantages**
- Facilitates reuse by providing code libraries
- Widely supported in browsers and server environments (*de facto* standard)

**Disadvantage**
- No means of "packaging" code (encapsulation)
ECMAScript Components...

• Are designed to fill this void.
• Builds upon the ECMA-262 standard.
Creating & Using Components

• Authors can “wrap” their ECMAScript code inside a component structure.
• Users can reuse the component inside of their Web pages or applications, working solely with its public interface.
Alternative Technologies

- Other scripting component technologies include Microsoft Scriptlets and Netscape JavaScript Beans.
- Key difference between ECMAScript Components and these others centers on their scope of the problem domain.
Alternative Technologies

- **Scriptlets** describe a scripting component that operates in a COM environment.
- **JavaScript Beans** describe a wider array of objects (e.g., HTML, Java), deal with integrating Java Beans and CORBA.
- **ECMAScript Components** focus squarely on the componentization of ECMAScript.
ECMAScript Component Interface

- Uses an XML vocabulary to describe the component to the outside world.
- Meta-information enclosed by a `<COMPONENT>` tag pair.

```xml
<COMPONENT NAME="MyComponent" SRC="SourceFile.js">
  <HELP URL="URL.HTML"/>
  <PROPERTY NAME="id"></PROPERTY>
  <METHOD NAME="execute"></METHOD>
  <EVENT NAME="onExecute"/>
</COMPONENT>
```
Design-Time Focus

• Normally for design-time usage, so it can be “detached” at runtime to simplify deployment
How a Tool Deals with Scripts

```<SCRIPT>
function ExcelControl(fN) {
    this.fileName=fN;
    this.graphIt=
        iCreateGraph;
    function iCreateGraph(){
        alert(""
    }
} Excel = new ExcelControl();
</SCRIPT>```

```<FORM>
<INPUT TYPE="button" NAME="OK" onClick="Excel.graphIt();">
</FORM>```

Without Componentization

Excel Control

**Action:**
Graph It!

**When:**
Click of OK Button

With Componentization
Technical Specification

- ECMAScript Component architecture
- XML vocabulary to define components
- Hosting ECMAScript Components
  - Generic
  - HTML document
For More Technical Info

- Richard Wagner, rw@netobjects.com
- Chip Shapley, chips@netscape.com