

1 ECMAScript Mobile Profile

ECMAScript Mobile Profile (ESMP) is a wireless scripting language that combines the most important features of JavaScript with support for low-bandwidth communication and thin clients. Using ESMP with XHTML Mobile Profile (XHTMLMP), you can give your mobile web clients more intelligence, creating a new class of products and services for ESMP-enabled phones.

This tutorial shows you how to use ESMP to dynamically control the behavior of site content. We'll use *event handlers*, pieces of ESMP code that are executed when a particular event occurs. We'll also manage images that appear on the screen through an image rollover effect.

1.1 The News Site Template

For this tutorial we'll design a site that gives users quick access to today's top stories and connects them to stories from the top news sources.

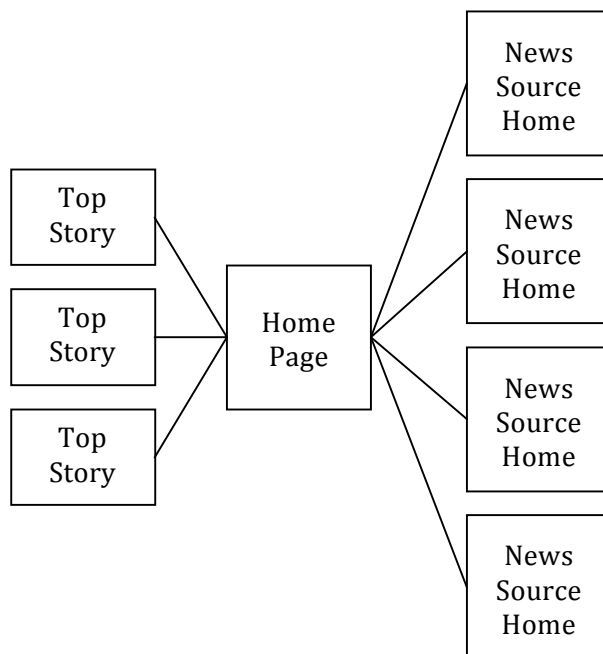
You could design a news site with simple XHTML, but we'll see how adding a little ESMP code can give you better control over the site and improve your users' experience.

1.2 Site Plan

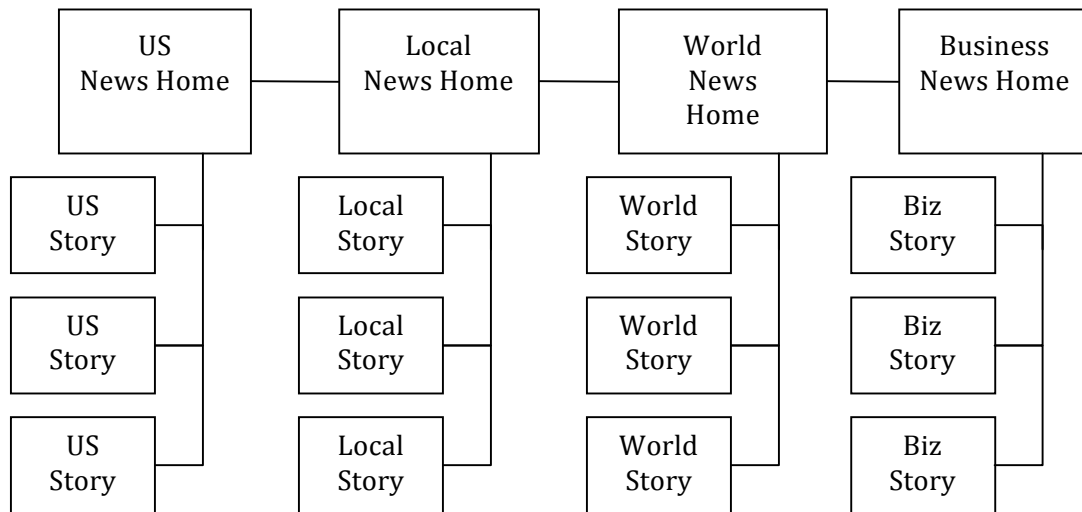
We can think of the site in two main sections:

- the home page and top story pages that link to it
- the news source sections and stories that link to them

We'll create a home page that links to three top story pages and to four selected news sources (CBS, ASG, BBC, and MSNBC):



We'll set up the news source sections to have multiple "home" pages for different types of news, each with links to related story pages:



Since we're designing a template for the site, we'll develop pages that can be used for multiple instances: for example we won't need to create a new design for each top story page. We want the look to be consistent across pages.

We'll create the home page first, followed by a top story page. Then we'll create the home pages for one of the news sources, ASG News, and look at how to design the ASG story pages.

2 Home Page

<i>Note</i>	You'll find the completed code for the home page for our news site in news.xhtml.
-------------	---

From the home page, users should be able to go to one of the day's top stories or to a favorite news source. We want the page layout to accommodate different screen sizes and to work well with the smaller real estate available on handset browsers.

Here's the design we choose, which shows the user the top headlines and lets them easily browse to the news source links:



2.1 Home Page Design: Nested Tables

First, before we worry about interactivity and ESMP code, we'll design the layout for the page. We can contain the main page design in two tables. We can design a table at the top of the page that holds the top margin, photo, and headlines.

If we add borders and exaggerate the spacing to show the structure of the table, it would look roughly like this:

News	
	Top Stories

The grid of links to news sources can be contained in a simple table:

The bottom table is fine as it is, but we need to add more to the top table to give us the layout we want. Into the dark blue cell in the top table, we insert another table to hold the photo and headlines:

--	--	--

And into the cell on the right of this second table, we insert a table with three rows, one for each headline:

These cells are slightly smaller than the dark blue cell behind them, so we see a dark blue border around them on the screen.

News	
	Top Stories

Here's the HTML code for the tables we've designed. Remember there's no ESMP code in the page yet—we'll add that in the next steps:

```
<table cellpadding="0" cellspacing="0" width="100%">

<!-- Top Margin -->
<tr><td bgcolor="white" colspan="2"></td>
</tr>
<tr><td width="100%" colspan="2"></td>
</tr>
```


2.2 Top Story Image Rollover

For our top three news stories, we have room for three short headlines, but only one photo. We show the first top story image when the page loads. Then, as the user scrolls down through the list of headlines, we'll use an image rollover effect to display corresponding images.

Let's add ESMP code to the three headlines. For each headline we register an event handler, `imageFocus()`, as the `onfocus` attribute of the link. We also add `accesskey` attributes so users can press number keys to jump to the linked pages.

```
<tr><td bgcolor="c3daec">
  <a href="top1.xhtml" onfocus="imageFocus(1);" accesskey="1">
    <u>1</u>&nbsp;&nbsp;&nbsp;Barbaro Wins Derby</a></td>
</tr>
<tr><td bgcolor="8db3d3">
  <a href="top2.xhtml" onfocus="imageFocus(2);" accesskey="2">
    <u>2</u>&nbsp;&nbsp;&nbsp;Rage In Sinai</a></td>
</tr>
<tr><td bgcolor="c3daec">
  <a href="top3.xhtml" onfocus="imageFocus(3);" accesskey="3" id="a3">
    <u>3</u>&nbsp;&nbsp;&nbsp;Kidnapped In Iraq</a></td>
</tr>
```

When the link gets focus, we'll use the image rollover effect in the `imageFocus` function to swap the displayed image. We use an XML DOM method, `getElementById()`, to retrieve a reference to the image we want to swap and then change its `src` attribute to change the image on the screen:

```
function imageFocus(num) {

  var img = document.getElementById("main_pic");

  switch (num) {          // change button
  case 1:
    img.src = "images/derby.png";
    break;
  case 2:
    img.src = "images/sinai.png";
    break;
  case 3:
    img.src = "images/kidnap.png";
    break;
  default:
    break;
  }
}
```

To be sure the photos come right up when we want them to, we define a function, `preload`, that initializes the three images:

```
function preload() {  
  
    var a, b, c;  
  
    a = new Image();  
    b = new Image();  
    c = new Image();  
  
    a.src = "images/derby.png";  
    b.src = "images/sinai.png";  
    c.src = "images/kidnap.png";  
}
```

We want to be sure this function is called as soon as the page loads, so we use the `onload` event handler registration to call the `preload()` function when the page loads:

```
<body onload="preload();" vlink="red">
```