macromedia®
FLASH™
enterprise kit for internet explorer 5.5

What is Macromedia Flash enterprise kit for Internet Explorer 5.5?

Using ViewLink, a new feature available in Microsoft® Internet Explorer 5.5, you can develop components in Dynamic HTML (DHTML) and Macromedia Flash (SWF). ViewLink enables encapsulation and rendering of content created in an HTML Components (HTC) file. Through the use of ViewLink, HTML, and Macromedia Flash content created inside an HTC file can render in the primary HTML document, thereby making the content of the document fragment visible to the user. ViewLink performs encapsulation by hiding the structure of the linked document fragment from the primary document.

ViewLink frees DHTML and Macromedia Flash authors to develop their own controls and other components in a language they are already familiar with, DHTML. ViewLink also makes components much easier to use because they are implemented as custom elements that can be used as simply as any other HTML tag. A ViewLink component requires no installation on the client machine, because it is downloaded and rendered as part of the primary document.

ViewLink and Macromedia Flash allow users to place pre-designed, parameterized Macromedia Flash content in the HTML pages, and define that behavior's content and appearance via information in the HTML file.

This kit provides examples and documentation on how to use Macromedia Flash with the new Internet Explorer 5.5 features.

Macromedia
“Microsoft Internet Explorer 5.5 provides Macromedia developers with a great open platform on which to build real-world, high-end Internet applications,” said Tom Hale, senior vice president of product marketing for Macromedia. “By working closely with Microsoft, Macromedia enables our combined developer communities to better take advantage of the Internet Explorer platform with powerful rich media.”

Microsoft
“Microsoft built Internet Explorer 5.5 so that third parties can build specialized components to meet their business needs,” said Chris Jones, vice president of the Windows Client Team at Microsoft. “We are excited to have Macromedia showcasing the ways developers can extend our platform.”

What is Macromedia Flash?

Macromedia Flash is the solution for producing and delivering high-impact Web sites, as well as resizable and extremely compact full-screen navigation interfaces, technical illustrations, long-form animations, and other dazzling site effects. High-quality viewing is assured, because graphics and animations anti-alias and scale are based on the viewer's screen size providing high-quality viewing. Macromedia Flash uses vector graphics technology. Unlike bitmapped images that are optimized for a single resolution, vector images can adapt to all the typical display sizes and resolutions. This is ideal for displaying Web sites uniformly on set-top boxes, hand-held computers, or PCs. Vector images–graphics, charts, maps, and animations–fit into compact, files that download more quickly and efficiently than bitmapped GIF and JPEG images.

Macromedia Flash content is created with the Macromedia Flash authoring software or Macromedia FreeHand® 9 print and Internet design solution. Macromedia Flash enables Web designers to import artwork from their favorite bitmap or illustration tools, apply transparency, create morphing effects, add interactivity and sound, and animate them over time. Flash content is then saved and published as a Flash Player file (also known as a SWF file) which is optimized for Web delivery.

What is ViewLink?

The ViewLink feature allows an element behavior to use DHTML for its user interface without affecting the document in which the behavior is used. To understand what this really means, let's examine a typical DHTML behavior in Internet Explorer 5.

If we take a look at the a custom element, <CAL:CALENDAR>, with a behavior attached that causes a fully dynamic calendar to render so that the user can select a date. The calendar UI is displayed using DHTML with a standard <TABLE> tag to lay out the grid of the calendar, and script to provide dynamic user functionality. To display the calendar, the table must be inserted into the main document using the property on the element. While this works, it has a number of disadvantages when it comes to building complex applications. The main document gets polluted with content that the original document author, using the <CAL> tag, was not aware of. For example, if a style rule defines all tables as pink with purple borders of 10 pixels, then the calendar will start to render in that fashion. The author of the document will think, "That's weird. I thought I just told only my tables to render that way, but it's affected my calendars too." For document authors, the fact that this component uses a table is purely an implementation detail, which they should not have to be aware of when doing their own authoring.

To prevent such fiascos, the ViewLink feature allows the author to fully encapsulate content to eliminate the possibility that the document structure of the primary document using the element behavior is going to be polluted. The behavior author no longer needs to worry about the document specifying styles that influence the behavior in unexpected ways.

So how does ViewLink work? By building on some features in Internet Explorer 5, where document fragments can be separate from the main document, we can now ViewLink a separate fragment so it can render for a particular element. This allows separate fragments of Macromedia Flash , or DHTML to be linked together for visual effect yet remain isolated from each other as far as the object model is concerned.

Implementing a ViewLink

There are two methods for setting up a ViewLink between a primary document and a document fragment−programmatic and declarative.
To programmatically set up a ViewLink between the primary document and a document fragment, the following statement is used. This statement is placed within script located in the HTC file.

```
defaults.viewLink=document;
```

A ViewLink can also be established in a declarative manner. The following example will automatically link the root element of the document fragment in the HTC file to the master element in the primary document.

```
<PUBLIC:DEFAULTS viewLinkContent/>
```

Primary Document and Namespaces

The ViewLink feature requires the use of XML namespaces in the primary document. You must declare a namespace in the primary document and prefix that namespace to your custom tags. Use the IMPORT processing instruction to associate the implementation of a ViewLink component with a namespace. The following example shows a primary document that uses a custom tag called TOOLBAR_BUTTON, which is implemented in a file called Toolbar_Button.htc.

```
 <HTML XMLNS:IE>

 <HEAD>

   <? IMPORT NAMESPACE="IE" IMPLEMENTATION="Toolbar_Button.htc">

 </HEAD>

 <BODY>

 <IE:TOOLBAR_BUTTON> </IE:TOOLBAR_BUTTON>

 </BODY>
</HTML>
```

The IE: declaration on the tag is required for the namespace of the custom tag to be declared. The IMPORT processing instruction is preceded with a question mark (?) to distinguish it as a processing instruction and not a regular element.

HTC File

The ViewLink code is located in an HTC file, which contains two sections: the tag and a tag. Here is a simple HTC file that uses declarative syntax to set the ViewLink on the object.

```
<!-- Toobar_Button.htc -->

<PUBLIC:COMPONENT tagName=TOOLBAR_BUTTON>

  <PUBLIC:DEFAULTS viewLinkContent/>

</PUBLIC:COMPONENT>



<BODY>

Some day this will be a toolbar button.
</BODY>
```

Examples

Please refer to the enclosed ViewLink behaviors folder for further documentation, source files, and examples.