

WebSlices:  
Windows® Internet Explorer® 8  
*Beta 1 for Developers*



*Making the Web Work for You*

March 2008

**For more information, press only:**  
Rapid Response Team  
Waggener Edstrom Worldwide  
(503) 443-7070  
[rrt@waggeneredstrom.com](mailto:rrt@waggeneredstrom.com)

The information contained in this document represents the current view of Microsoft Corp. on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording or otherwise), or for any purpose, without the express written permission of Microsoft Corp.

Microsoft may have patents, patent applications, trademarks, copyrights or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place or event is intended or should be inferred.

© 2008 Microsoft Corp. All rights reserved.

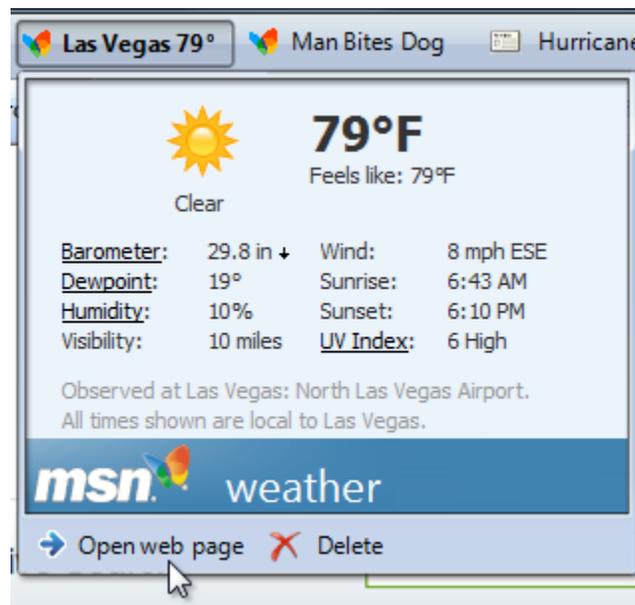
Microsoft, Windows, Windows Vista, Windows Server, ActiveX, Active Directory, Internet Explorer, the Internet Explorer logo, MSN and the MSN logo are either registered trademarks or trademarks of Microsoft Corp. in the United States and/or other countries. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

## OVERVIEW

WebSlices allow users to subscribe to portions of a Web page.

Users commonly visit many Web sites several times a day to see if there are new updates. RSS feeds make this experience easier: Users can subscribe to updates and view summaries of them without having to navigate to each site individually. Windows Internet Explorer 8 Beta 1 for Developers supports subscribing to feeds today and adds support for WebSlices.

WebSlices are a new feature for Web sites to connect with users by allowing users to subscribe to content directly within a Web page. WebSlices behave just as feeds do, where clients can subscribe to get updates and notification of changes.



A WebSlice is a portion within a Web page that is treated as an item that can be subscribed to, just as a feed can. To enable WebSlices on your Web site, annotate your Web page with class names for the title, description, and other properties that can be subscribed to.

Internet Explorer 8 Beta 1 for Developers users can discover WebSlices within a Web page and add them to the Favorites Bar, a dedicated row below the Address Bar for easy access to links. Internet Explorer 8 Beta 1 for Developers subscribes to the Web page, detects changes in the WebSlice, and notifies users of updates. Users can preview these updates directly from the Favorites Bar and click through to the Web site for more information.

## FEATURE DETAILS

This document covers the details of the WebSlice feature.

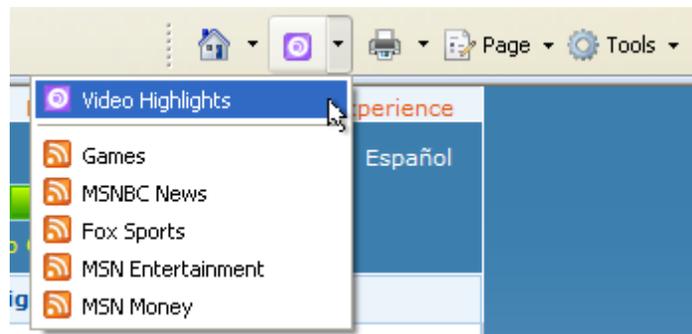
1. [User Interface](#)
2. [WebSlice Authoring](#)
3. [Discovery of WebSlices](#)
4. [Client Processing of WebSlices](#)

### 1. User Interface

There are four aspects to the user experience of WebSlices. Please note that the following visuals are not final.

- Discovery
- Subscribe
- Notify
- Preview

**Discovery:** Users can discover that a Web page has WebSlices, just as the page could have feeds. The discovery button lights up if there are WebSlices or feeds found on a current Web page.



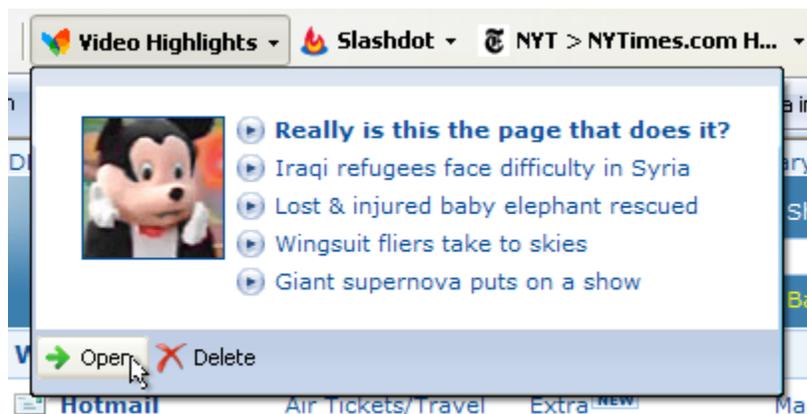
Users can also discover WebSlices directly within a Web page: When a mouse pointer hovers over a WebSlice region, an in-page button appears at the top left corner.

**Subscribe:** Users can add WebSlices to the Favorites Bar in order to get updates, by clicking on the Discovery button.

**Notify:** Users can see when there are updates to the WebSlices. WebSlices that have updates are bold on the Favorites Bar.



**Preview:** Users can preview updates to WebSlices. Clicking a WebSlice on the Favorites Bar displays a details view of that WebSlice. Users can navigate to the Web site that hosts the WebSlice by clicking on the open button.



HTML displayed in the details view comes from the description property of a WebSlice. The preview window has limited ability; for example, the window allows no scripting or form controls. Also, links within the view navigate using the current tab.

You can use WebSlices to keep users connected to your Web site. With WebSlices, users get updates, and have an easy link back to your Web site. WebSlices can be used for weather, horoscopes, and "what's new" information, as well as time-dependent content such as auction items and flight arrivals. The next section covers how to enable your site with WebSlices.

## 2. WebSlice Authoring

WebSlices are enabled by adding HTML annotations directly to the Web page. WebSlices use a combination of the [hAtom Microformat](#) and the WebSlice format to describe a subscribable portion of a Web page. This section covers the primary, expiration, and bandwidth properties of a WebSlice.

### The Basics

The basic properties of a WebSlice are the WebSlice container, Entry Title, and Entry Content.

#### WebSlice

- Required.
- Uses class name **hslice**.
- The WebSlice element must include an ID.
- Represents the entire region of a WebSlice that contains the title, description, and other WebSlice properties.

#### Entry Title

- Required.
- Uses class name **entry-title** as defined in the [hAtom Microformat](#).
- There can be multiple instances of Entry Title.

#### Entry Content

- Optional.
- Uses class name **entry-content** as defined in the [hAtom Microformat](#).
- If the Entry Content is missing, assume it is the empty string.
- There can be multiple instances of Entry Content.

The following is an example of how to mark up an auction item with the link, title, and description properties.

**<http://auction.microsoft.com/item>**

```
<div class="hslice" id="1">
  <p class="entry-title">Item - $66.00</p>
  <div class="entry-content">high bidder: buyer1
  ...
</div>
</div>
```

Based on the markup, the WebSlice link is <http://auction.microsoft.com/item#1> and the title is "Item - \$66.00."

## Expiration

WebSlices also handle content that is dependent on time; for example, an auction item ends tomorrow or a flight arrives at 7 P.M. today. WebSlices are active until the expiration is reached, and warn when they are about to expire.

### End Time

- Optional.
- Uses class name **endtime**.
- Uses the [abbr-design-pattern](#) to encode the date and time.

In this case below, the WebSlice expires on February 28 at 5 P.M.

```
<div class="hslice" id="1">
  <p class="entry-title">Item - $66.00</p>
  <div class="entry-content">high bidder: buyer1
    ...
    <p>End time:<abbr class="endtime" title="2008-02-
28T17:00:00">1 day 18 hours</abbr></p>
  </div>
</div>
```

If a user has Internet Explorer 8 Beta 1 for Developers open, the WebSlice button indicates expiration by glowing around 5 minutes before 5 P.M. After 5 P.M. the text on the button is gray, indicating that the WebSlice has expired. The button is still active in the expired state, so that users can navigate to the Web page.

## Bandwidth

Subscription to Web pages add additional load on servers and hence are a concern to publishers. In order for publishers to control how frequently clients check for updates, WebSlices can include a ttl (Time-to-Live) property that indicates the minimum refresh interval. The value means that Internet Explorer 8 Beta 1 for Developers won't check for updates more frequently than the defined value. For more details, see the description of ttl in the RSS Platform [blog post](#).

For example, if a WebSlice specifies a ttl value of 180 minutes, the download engine will not check for updates more frequently than every 3 hours, even if the user has set the scheduled interval to 60 minutes.

### Time to Live

- Optional.
- Uses class name **ttl**.
- Represents the concept of [RSS 2.0 ttl](#).

- The value of the Time-to-Live element must be an integer.
- Indicates the number of minutes that a client should cache the data.

```
<div class="hslice" id="1">
  <p class="entry-title">Item - $66.00</p>
  <div class="entry-content">high bidder: buyer1
    ...
    <p>This clip updates every <span class="ttl">15</span>
    minutes </p>
  </div>
</div>
```

Alternatively, you can specify a different source that clients should get updates from. If your Web page has a lot of content that is unrelated to a WebSlice, you can redirect clients to subscribe to an RSS feed. The alternative feed contains a single feed item with all of the same properties as the WebSlice.

### Feed URL

- Optional.
- Uses rel name **feedurl**.
- Represents the alternative feed to which clients should subscribe for updates.
- The URL of the alternative feed is specified as the value of the **ref** attribute.

```
<div class="hslice" id="1">
  <p class="entry-title">Item - $66.00</p>
  <div class="entry-content">high bidder: buyer1
    ...
    <a rel="feedurl"
      href="auction.microsoft.com/item.xml">Subscribe to
      WebSlice</a>
  </div>
</div>
```

The referenced feed should be a single-item feed. The feed item represents the WebSlice. Use the **mon** namespace if you want to specify the expiration (endtime) for the WebSlice, as shown in green.

```
<?xml version="1.0"?>
<rss
  version="2.0" xmlns:mon="http://www.microsoft.com/schemas/rss/monitoring/2007">
  <channel>
    <title>Feed for Item</title>
    <link>http://auction.microsoft.com</link>
    <description>Feed for auction item</description>
    <ttl>15</ttl>

    <item>
      <title>Item = $66.00</title>
      <link>http://auction.microsoft.com/item#1</link>
      <description>high bidder: buyer1 ...</description>
      <mon:endtime>2008-02-28T17:00:00</mon:endtime>
    </item>
```

```
        </channel>
    </rss>
```

When the alternative source property is present, Internet Explorer 8 Beta 1 for Developers subscribes to the specified feed to get updates. The original Web page is no longer used for getting updates to the WebSlice.

### WebSlice Details View ( HTML)

The outer HTML extracted from the entry-content element is used for display in the details view. The styles associated with the HTML of the entry-content element and the original webpage are used:

- imported stylesheets
- style rules defined in head element
- direct in-line styles

Styles from the parent element do not get applied in the HTML of the details view. In this example, the details does not display the blue background because the style is applied to a parent element and not the entry-content element.

```
<div class="weather" style="background:blue">
  <div class="hslice" id="main">
    <h2 class="entry-title">Seattle Weather</h2>
    <p class="entry-content">The weather is X.</p>
  </div>
</div>
```

To apply the style to the HTML flyout, include the style information to the entry-content element:

```
<div class="weather" style="background:blue">
  <div class="hslice" id="main">
    <h2 class="entry-title">Seattle Weather</h2>
    <p class="entry-content" style="background:blue">The weather is
X.</p>
  </div>
</div>
```

When rendering the details view, the body element is inserted around the outer HTML of the entry-content element. If you have a stylesheet that applies to the body element, it will also be applied within the body of the HTML flyout. To avoid having the body style applied to the HTML flyout, you can specify a class name to the body of the original Web page where the styling references the class name. In the

example below, the green background applied to original web page will not be used in the HTML details view.

```
<style type="text/css">
    .MyBody { Background-color: Green };
</style>

<body class="MyBody">
    <div class="weather">
        <div class="hslice" id="main" style="background:blue">
            <h2 class="entry-title">Seattle Weather</h2>
            <p class="entry-content">The weather is X.</p>
        </div>
    </div>
</body>
```

You can also specify an alternative feed, where the item description contains the direct styling for the HTML details view.

```
<item>
    <title>Seattle Weather</title>
    <link>http://www.msn.com</link>
    <description> <![CDATA [
        <p style="background:blue">The weather is X.</p>
        ...
    ]]>
</description>
</item>
```

## Authenticated WebSlices

There are two ways to authenticate WebSlices: persistent cookies and HTTP Authentication.

### Persistent Cookies

Cookies that are enabled for a Web page are used as part of the background update. When the cookies are expired, it is recommended to send a link to login as part of the WebSlice entry-content property. This allows the user to get back to your site and reset the cookies.

There is a known issue with cookies not properly passed for background updates on Windows Vista® with Protected Mode on. To work around this issue, users must run Internet Explorer 8

Beta 1 for Developers in elevated mode (right-click the Internet Explorer 8 icon and select **Run as administrator**) in order to get the authenticated WebSlice content.

### HTTP Authentication

Internet Explorer 8 Beta 1 for Developers supports WebSlices that are authenticated via HTTP authentication. Specifically, HTTP BASIC and HTTP DIGEST authentication are supported (same as any regular web page). However, for WebSlices HTTP BASIC authentication is **only** supported via HTTPS (SSL).

WebSlice support	HTTP	HTTPS
HTTP BASIC auth	-	✓
HTTP DIGEST auth	✓	✓

Let's use HTTP BASIC auth via SSL for this example. Make sure that when the users are browsing to a WebSlice, that they are doing so via HTTPS so that when the user subscribes to the WebSlice they are subscribing to it using HTTPS.

The server must challenge the requests to the WebSlice with a regular HTTP auth challenge (HTTP 401 Auth Required response). Initially, after the user subscribes, they will not have saved the username and password with the WebSlice subscription. However, the first time they view the WebSlice, they will be informed that the WebSlice requires authentication and they should enter their credentials into the WebSlice properties dialog. Once entered, the WebSlice will be correctly updated going forward.

### Using an alternate URL to subscribe to an authenticated WebSlice

It is possible to make the WebSlice reachable via HTTP for browsing purposes and HTTPS after subscribing. Include an alternate URL (<a rel="feedurl" href="https://...."> construct) to subscribe to the WebSlice. In this case the feedurl should point to the same WebSlice but instead of HTTP it should be HTTPS so that the user will be subscribed to the SSL version.

This approach has the additional benefit of allowing a regular HTTP WebSlice to use cookie authentication and the HTTPS version to use HTTP Authentication. This means that HTTP Authentication is only used when the user is actually subscribing to a WebSlice and not when just browsing to pages with WebSlices.

### 3. Discovery of WebSlices

An earlier section discussed the experience of discovering WebSlices within a Web page. Just by adding WebSlice annotations to your Web page, you direct Internet Explorer 8 Beta 1 for Developers to automatically detect the presence of WebSlices, and to light up the discovery button and the in-page button.

This section covers the details of how WebSlice discovery works, along with additional options that let you make this experience better for your site. You can specify the main WebSlice for a Web page, disable the In-page button, and promote WebSlices directly within your Web page.

#### WebSlice Discovery

The discovery button lists the discovered WebSlice in the order that Internet Explorer 8 Beta 1 for Developers parsed the document. The discovery button is a split button, where the default action goes to the first discovered WebSlice. If you want to specify the main WebSlice for the page, do so by adding a link element in the head of the document:

**rel Attribute:** Required. The value must be "default-slice".

**type Attribute:** Required. The value must be "application/x-hatom".

**href Attribute:** Required. This is the URI for the clip within the current document. The URI will be ignored if it is outside of the current document.

```
<head>
  <link
    rel="default-slice"
    type="application/x-hatom"
    href="www.example.com/#auction" />
</head>
```

Specifying a main WebSlice for a document sets the default action of the discovery button to subscribe to the main WebSlice. It will also automatically set the default **Add to Favorites Bar** menu item to add the WebSlice.

#### In-Page WebSlice Discovery

When pointer hovers over a WebSlice region—the HTML element that contains the **hSlice** class name—the in-page WebSlice appears at the top left corner of the window. This provides greater discoverability of the WebSlice.

If you wish to have this capability turned off for a particular Web page, add the following metadata to the head of the page:

**name Attribute:** Required. This indicates that the META tag is related to WebSlice functionality. The value must be "slice" (case insensitive).

**scheme Attribute:** Required. The value must be "IE" (case insensitive).

**content Attribute:** Required. This indicates that the in-page button should be turned off for the current page. The value must be "off" (case insensitive).

```
<head>
  <meta name="slice" scheme="IE" content="off" />
</head>
```

When you have this capability disabled, you will need to advertise how users can add WebSlices. This can be done through the **AddToFavoritesBar** API method.

### AddToFavoritesBar API Method

This is a new method for Web sites to promote adding to the Favorites Bar. When the user invokes this method, the dialog allowing subscription to a WebSlice appears.

```
window.external.addToFavoritesBar(string URL, string Title, [optional]
string Type)
```

**URL:** Required. This is the URL of the WebSlice to add. It is the full URL that includes the anchor of the WebSlice.

**Title:** Required. This is the name of the WebSlice.

**Type:** Optional. This is the type of link. This value must be "slice" in order for Internet Explorer 8 Beta for Developers to subscribe to this WebSlice.

This is an example of how to implement this API for a button within a Web page.

```
<button
onclick="javascript:window.external.addToFavoritesBar('http://auction.microso
ft.com/item#1', 'Item - $66.00', 'slice')">Add WebSlice</button>
```

### Refresh Parsing

If a client application dynamically updates **link** elements for search providers and feeds, it can call [window.external.contentDiscoveryReset\(\)](#) to parse the document again. This action causes Internet Explorer 8 Beta 1 for Developers to reload the list of feeds, WebSlices, and search providers on the **Feed Discovery** button and **Instant Search** drop-down list, which reflect the new document state.

Refresh parsing can be used in powerful ways; for example, a map site could provide feeds for all the theaters currently in the map view without refreshing the page. This works well for content that is used immediately while viewing the page, but it can be tricky when working with content that is downloaded later after page load, such as a WebSlice. Some sites may need the newly downloaded content, such as a WebSlice or feed, to be newly discovered. If the WebSlice is inserted into the page dynamically after navigation, you must provide a link to an alternate feed with [rel=feedurl](#) in order to get updates to that item.

## 4. Client Processing of WebSlices

Internet Explorer 8 Beta 1 for Developers treats WebSlices just as it does feeds: The WebSlice is added to the Common Feed List to be handled by the Windows Feeds Platform. This makes WebSlices available within not only Internet Explorer 8 Beta 1 for Developers, but also other applications on Windows.

The Windows Feeds Platform updates feeds and WebSlices at a regular frequency specified by the user. By default, the frequency is once per day. Users can increase this to as often as every 15 minutes through the WebSlice **Properties** dialog.

During the update process, the Windows Feeds Platform passes feeds and WebSlices through a sanitization process; among other things, this process removes script code from the description element. This mitigates the effect of any malicious content on Internet Explorer 8 Beta 1 for Developers and other applications using the platform. At this step, the WebSlice from the server is compared to the WebSlice stored locally. If the titles or descriptions of the WebSlices differ, the item is marked as updated and appears as a bold button on the Favorites Bar.

Because of WebSlice sanitization, any reference to script or HTML form fields will not work within the details view of the WebSlice button. As an additional precaution, the details view is set to the restricted zone where script code does not run.

## 5. Known Issues

These are the known issues in Internet Explorer 8 Beta 1 for Developers:

- If initial size of the WebSlice details view uses the size of the entry-content rectangle discovered from the original Web page. If the size appears differently in the details view, try setting the Web page to render in IE7 Standards mode.
- Cookies do not work properly in Windows Vista with Protected Mode on. To work around this issue for users on Windows Vista, you can redirect users to a HTTPS feed as specified in the "Using an alternate URL to subscribe to an authenticated WebSlice" section.