

## **AJAX Best Practices - Becky Gibson**

From <http://www-3.ibm.com/able/resources/ajaxaccessibility.html>

### **See also the Dynamic Accessible Web Content Road Map:**

<http://www.w3.org/WAI/PF/roadmap/DHTMLRoadmap040506.html>

Until the use of AJAX is ubiquitous, it is important to inform the user about the requirements of an AJAX application. This section describes some current best practices for AJAX usage.

#### **Provide an indication that scripting is required**

Because RIAs rely on scripting, you should indicate on an entry page in the application that scripting is required. Provide a statement early in the page that indicates the use of AJAX technologies and informs the user that dynamic updates will occur.

#### **Provide an option for manual notification**

Some applications provide automatic updates of data without explicit request by the user. Examples include weather, stock, or traffic updates. When building these types of applications consider offering an option to disable the automatic update and allow the user to manually request the updates. Save the user's update preferences on the server for applications that require authentication or by using browser cookies.

#### **Provide notification of updates**

If the site relies on asynchronous updates to the page, consider giving the user the option of receiving an alert when an update occurs. This will help notify those users of AT who can't see that an update to the screen has occurred. Providing an alert when change occurs also helps screen-magnifiers users whose focus could be scrolled away from the changed area. And, alerts assist people who may not easily notice the change due to cognitive or attention issues. This should be an optional notification that the user must elect to enable because the alert itself may prove distracting for some users.

#### **Provide notification of focus shift**

Do not automatically shift focus on the page when an update occurs. Changing focus without warning can be distracting for some users, especially if there is no easy mechanism to return to the previous position. For example, on a site that dynamically updates a series of stock prices located at the bottom of the page, a sudden shift in focus each time a price change occurs could make the page unusable.

#### **Consider the use of color or a change in font size or weight to temporarily highlight the area which has updated.**

Just like notification of updates, changes in color must be an elective option. A person using a screen reader or magnifier may want notification by means of an alert or focus change, while a sighted user may prefer a color change. When changing background color, it is best to use a subtle color and only change the background for 5-10 seconds. But, be cautious of causing a dramatic change or blinking that might distract some users.

Some changes in focus, however, may be appropriate. If the user has activated a button to check for new mail, it may be appropriate to move focus to the list of mail messages when the update has completed. Make certain that the user is aware that an action is going to occur and expects focus to change to the results.

#### **Make navigation easy**

Provide links to portions of the page that are dynamically updated. If the site relies on the update of several areas of the page at different frequencies, provide a way to quickly navigate to each section. This set of links should be easily reached from the top or bottom of the page to make navigation to them quick and easily repeatable.

Proper use of semantic markup is also important. Assistive technologies provide a mechanism for navigating by headers. Using the HTML header element to mark sections within the page allows assistive technology users to navigate from section to section and find updated content. Consider a search form that uses AJAX to retrieve the search results rather than reloading the entire page. If the section where the search results are returned is coded with an HTML header element labeled "Search Results," assistive technology users can easily navigate to the updated results section after submitting the search request. This is good HTML coding practice, which is essential in the world of AJAX-enabled Web pages.

### **Update elements with new content instead of creating new elements**

Whenever possible, update existing elements with new content rather than creating and adding new high-level elements to the page. If you must add elements, append them after the current focus point and make them a parent to an existing element rather than the document itself. For example, create a <div> element in the markup and use it as the parent element when you add new content to the page. Update the contents within that <div> but don't remove and recreate the parent <div> element itself.

### **Notes:**

Adding navigable elements with explicit tabindex attribute values within the page may affect the tab order for AT users.

Various ATs handle dynamic additions to a page differently, so testing with AT is essential to ensure a fully-accessible page.