YUI Library: SWFStore

2011-3-21 **v2.9**

Instantiating the SWFStore

```
<div id="SWFStoreDiv" style="width:0px;height:0px;">
</div>

<script>
var swfstore = new YAHOO.util.SWFStore("SWFStoreDiv");
</script>
```

Instantiates a new SWFStore object, `swfstore`, which is bound to a div whose id attribute is `'SWFStoreDiv'`. This will create an invisible component that won't take up any space. To make the component large enough to display user settings, simply set the size of the div to `height:215; width:138;`

Constructor

```
YAHOO.util.SWFStore(str element, bool shareData,
   bool useCompression );
```

Arguments:

(1) **element:** HTML ID for the SWFStore container. May be empty or contain alternative content. Size and background color will propagate to SWF
(2) **shareData:** Whether to share data across browsers. *(optional)*
(3) **useCompression:** Whether to compress data when stored. *(optional)*

Dependencies

SWFStore requires the Yahoo Global Object, Dom, Cookie, Event and Element, as well as SWF and SWFDetect utilities. The SWFStore also uses the `swfstore.swf` file that must reside in the same path as the page. On the client side, SWFStore requires that the user have **Flash 9.0.115** or later installed in their browser, and have not turned off local storage for Flash Player.

Security Considerations

By default, SWFStore uses a whitelist to determine which pages are allowed to load the swfstore.swf file and manipulate storage. To use this, create an XML file called storage-whitelist.xml in the same directory as the swfstore.swf file. Include any number of allow-access-from nodes, which point to a full URL. Any URL specified here will be allowed access. Be as specific as necessary. For instance, specifying http://www.yahoo.com will allow http://www.yahoo.com/mail and http://www.yahoo.com/preferences. However, only specifying http://www.yahoo.com/preferences would not allow http://www.yahoo.com.

```
<?xml version="1.0" encoding="utf-8"?>
<url-policy>
       <allow-access-from url="http://www.yahoo.com" />
       <allow-access-from url="http://www.yahoo.com/mail" />
</url-policy>
```

Simple Use Case

```
swfstore.addEventListener("save", onSuccess);
function onSuccess (event) {
   alert("Your username has been stored");
}
swfstore.setItem('my_username', 'Jed90210');
```

Creates a store for a username under the location "my_username". When the item is successfully stored, an alert will pop up.

Considerations

(1) By default, SWFStore looks for `swfstore.swf` in the same directory as the calling page, and this is not currently configurable. Therefore, you must host the swf on your own server.
(2) SWFStore genrally allows 100kb of storage for a particular domain, by default. If you expect your storage to go above the 100kb limit, an advance call to `setSize()` and notification of your users is recommended.
(3) The local storage files are **unencrypted** binary files, easily accessible in a specific folder on the user's system. Storing user-identifiable, or private data is therefore not recommended. Note that clearing browser cookies does not remove this data.

Solutions

Requesting more storage

To request storage over 100kb, a Flash dialog will pop up in the SWFStore SWF. If your container element is sized to not show the SWFStore, it will need to be resized if more storage is requested. If the current size of the SWF is not at least 215px wide and 138px tall, the dialog will not be able to display and the attempted store will fail.

```
SWFStoreDiv.style.width = "215px";
SWFStoreDiv.style.height = "138px";
```

The following requests 1mb of storage for the domain:

```
swfstore.setSize(1000);
```

If an item is attempted to be stored that is larger than the current available size, this settings dialog will display automatically.

Removing all data from storage

If you would like to clear all data from a particular store:

```
swfstore.clear();
```

YAHOO.util.SWFStore Events:

success
error
pending
openDialog
openExternalDialog

YAHOO.util.SWFStore Methods:

calculateCurrentSize()
clear()
displaySettings()
getItems()
getNameAt(index)
getLength()
getModificationDate()
getShareData()
setShareData()
getTypeOf(str)
getTypeAt(int)
getUseCompression()
getIValueOf(str)
getValueAt(int)
removeItem(str)
removeItemAt(int)
setItem(str, obj)
setSize(int)
setUseCompression()