The Yahoo! Developer Network

- Utility and Data Web Services
- Design Patterns Library
- Browser Support Guidelines
- Yahoo! User Interface Library (YUI)
The Yahoo! Developer Network Ecosystem: Utility and Data Web Services

Services

- Browser Based Authentication (BBAuth), Answers, Local, Maps, Photos, Search, Shopping, Travel, Utilities, (Mail coming soon)

Hackable Sites

- del.icio.us, Flickr, Upcoming.org, Webjay

Burnable Feeds

- Finance, HotJobs, RSS Feeds, Traffic, Weather

SDKs

- Messenger, Music, Search Developer Kit, Widgets

Developer Centers

- JavaScript, Flash, .NET, PHP, Python, Ruby
Welcome
Welcome to the Yahoo! Design Pattern Library. You're looking at the second of many periodic releases, many of which include pointers to the Yahoo! User Interface Library code. We're thrilled to be sharing patterns and code with the design and development community, hope it's useful, and look forward to your feedback.

What's a Pattern?
A pattern describes an optimal solution to a common problem within a specific context. more...

Recent Patterns see all...

Animate Transition
Designer needs to communicate that an object is changing its spatial relationship within the page.

Collapse Transition
The designer needs to communicate that an object is no longer of primary importance, yet wants to keep it available in a smaller.

Dim Transition
Designer needs to communicate that an object is of secondary importance.
The Yahoo! Developer Network Ecosystem: Design Patterns

Design Pattern Library

Yahoo! Developer Network > Design Pattern Library > Auto Complete

Auto Complete

Problem Summary
The user needs to enter an item into a text box which could ambiguous or hard to remember and therefore has the potential to be mis-typed.

Example:

To: yu
Cc: Yusef Jones <yusef@somewhere.com>
Yusef Smith <yusef@someplace.org>
Subject: yui_blogger <yui_blogger@yahoo.com>

Hello,

Auto completion of contacts in Yahoo! Mail Beta

Quick Jump
- Solution
- Rationale
- Accessibility

Related Patterns
- Button Sequence
- Chooser
- Input Guidance

As Used On Yahoo!
- Yahoo! Mail Beta
- Yahoo! Maps Beta
- Yahoo! My Web 2

Blog
- Blog Article

YUI Code Examples
- Auto Complete Component
The Yahoo! Developer Network Ecosystem:
Design Patterns

Hello.

Auto completion of contacts in Yahoo! Mail Beta

Use When

- The suggestions can be pulled from a manageable set of data.
- The input item can be entered in multiple ways.
- The input item can be matched with a specific data item in the system.
- Speed and accuracy of entry is an important goal.
- The total number of items would be too large or inconvenient for displaying in a standard drop down box.

Solution

Layout

- Use a standard text box for input.
- Label the text box to match the user’s expectation of what field will be searched against.

Interaction

- As the user types, display a list of suggested items that most closely match what the user has typed. Continue to narrow or broaden the list of suggested items based on the user’s input.
- Display the suggested items list in a drop down box directly underneath the text box. The suggested items list may be based on the complete set of data or more narrowly based on other criteria such as each item’s frequency of use.
- When available, show multiple fields of information for each suggested item. In the
Graded Browser Support

By Nate Koechley
Senior Front-End Engineer
Yahoo! Inc.
February 13, 2006

In the first 10 years of professional web development, back in the early '90s, browser support was binary: Do you — or don’t you — support a given browser? When the answer was “No”, user access to the site was often actively prevented. In the years following IE5’s release in 1998, professional web designers and developers have become accustomed to asking at the outset of any new undertaking, “Do I have to support Netscape 4.x browsers for this project?”

By contrast, in modern web development we must support all browsers. Choosing to exclude a segment of users is inappropriate, and, with a “Graded Browser Support” strategy, unnecessary.

Graded Browser Support offers two fundamental ideas:

- A broader and more reasonable definition of “support.”
- The notion of “grades” of support.

What Does “Support” Mean?

Support does not mean that everybody gets the same thing. Expecting two users using different browser software to have an identical experience fails to embrace or acknowledge the heterogeneous essence of the Web. In fact, requiring the same experience for all users creates a barrier to participation. Availability and accessibility of content should be our key priority.
The Yahoo! Developer Network Ecosystem: Browser Support Guidelines

A-Grade Browser Support Chart

This chart lists all browsers that receive A-Grade support as defined by [Graded Browser Support](#).

<table>
<thead>
<tr>
<th></th>
<th>Win 98</th>
<th>Win 2000</th>
<th>Win XP</th>
<th>Mac 10.3.x</th>
<th>Mac 10.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE 7.0</td>
<td>A-grade</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IE 6.0</td>
<td>A-grade</td>
<td>A-grade</td>
<td>A-grade</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Firefox 2.0.*</td>
<td>A-grade</td>
<td>A-grade</td>
<td>A-grade</td>
<td>A-grade</td>
<td>A-grade</td>
</tr>
<tr>
<td>Firefox 1.5.*</td>
<td>A-grade</td>
<td>A-grade</td>
<td>A-grade</td>
<td>A-grade</td>
<td>A-grade</td>
</tr>
<tr>
<td>Opera 9.*</td>
<td>A-grade</td>
<td>A-grade</td>
<td>A-grade</td>
<td>A-grade</td>
<td>A-grade</td>
</tr>
<tr>
<td>Safari 2.0*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A-grade</td>
</tr>
</tbody>
</table>

Changes

Primary Changes

1. REITERATION of A-Grade support, IE 7, Win
2. TERMINATION of A-Grade support, IE 5.5, Win
3. INITIATION of A-Grade support, Firefox 2.x, Win/Mac
4. TERMINATION of A-Grade support, Firefox 1.0 x, Win/Mac
Yahoo! User Interface Library

The Yahoo! User Interface (YUI) Library is a set of utilities and controls, written in JavaScript, for building richly interactive web applications using techniques such as DOM scripting, DHTML and AJAX. The YUI Library also includes several core CSS resources. All components in the YUI Library have been released as open source under a BSD license and are free for all uses. Download the entire library, its documentation and examples from Sourceforge.

The library's developers blog at the YUI Blog and the YUI Library community exchanges ideas at YDN-JavaScript on Yahoo! Groups.

YUI Theater:
Matt Sweeney, "Web 2.0: Getting It Right the Second Time"

YUI developer Matt Sweeney delivers a talk on frontend architecture for Web 2.0 at Yahoo's Open Hack Day. (Visit Yahoo Video for the full-format version.)
Why?
changing seasons

on the web

http://flickr.com/photos/getthebubbles/107463768/
People expect less online.
But we are online...
... and tied to the browser.
So we must level the playing field.

http://www.flickr.com/photos/probek/44480413/
it takes 2 things

http://flickr.com/photos/latitudes/104286031/
We must improve our technology
Specifying
Implementation
Defects
[ Theory / Practice ]
knowledge areas: 7
dimensions: x 4
platforms: x 3
browsers per platform: x 4
rendering modes: x 2

= 672
(we need a robust platform!)
and raise expectations
(so users engage and explore with confidence!)
from Implementation Models

design

to Mental Models
from Heterogeneous Environments to Compelling and Consistent APIs
Nuts and Bolts

http://flickr.com/photos/snood/129758197/
Six commitments to the platform
1] Meet properties where they are today. Facilitate incremental enhancement (“Transitional Internet Applications”)
2] Provide a broad, inclusive platform for Web 2.0’s “Rich Internet Application” development (a la carte, not framework)
3] Extensible and adaptive, to meet the needs of our diverse product portfolio (lots of different contexts)
4] Support all A-grade browsers – an evolving and still demanding challenge
(Graded Browser Support)

5] Support our scale and scope
(industrial grade)
6] Be responsive and accountable to the community of designers and engineers
(your P1 bugs are our P1 bugs)
Learn from Desktop History

Adapted from Alan Cooper’s “About Face 2.0” Book

- **Idioms**: Property-specific, application-level UI innovations
  - Slider, calendar, autocomplete, treeview

- **Widgets**: Generic input/output interaction patterns for common tasks
  - Event management, drag and drop, connection manager, animation
  - Smarter, faster, network-wide interfaces

- **Utilities**: Foundational tools for processing browser primitives
  - Event-sensitive, in-page rich internet applications

- **Platform**
- **Property**
The Yahoo! User Interface (YUI) Library

- TreeView
- AutoComplete
- Slider
- Calendar Control
- TabView
- Menu Control
- Logger Control
- DHTML Windowing
- Animation
- Drag & Drop
- Connection Manager (Ajax)
- Event Utility
- Dom Collection
- Loader
- CSS Reset, Fonts, Grids
YUI DOM Collection

- `isAncestor(parent, potentialChild)`
- `inDocument(el)`
- `getDocumentHeight, getDocumentWidth`
- `getViewportHeight, getViewportWidth`
- `getElementsBy`
  - `YAHOO.util.Dom.getElementsBy(function(el) { return (/^http:\/\/www\yahoo\./.test(el.getAttribute('href'))); }, 'a', 'content'))`;
- `Class management utilities`
• Event
  – Flexible batch assignment
  – onAvailable and onContentReady
  – Scope correction and assignment
  – Automatic cleanup
  – Custom Events
YUI Event’s Custom Events

- `var myEvent = new YAHOO.util.CustomEvent('myEvent');`
- `myEvent.subscribe(function() {
    alert('event fired');
});`
- `myEvent.fire();`
YUI Animation

```javascript
var anim = new YAHOO.util.Anim(el, {
    width: {to: 400},
    height: {by: 400, unit: 'em' },
    opacity: {from: 0, to: 1}
});

anim.animate();
```

- Includes support for Bezier math control
• **File uploading** in addition to XHR
  – Easy implementation of file uploading across the A-Grade browsers
  – Use the upload member of the callback object (instead of "success" or "failure")

• XHR callback with extra arguments
Drag and Drop

• It’s easy, sorta, but easy to get wrong too
• Point, Region, Multiple Handles, !Handles
• Exposes 15 interesting moments
What happens when the mouse is pressed on the draggable object but dragging has not initiated?
YUI Logger and Firebug

- alert("stop the insanity");
- YAHOO.log("There IS a better way");
- Debug with Logger or Firebug directly.
CSS Grids

- Page Widths
- Template Presets
- Nesting Grids

- Together, 200 layouts for 2kb.
http://yuiblog.com/blog/2006/10/16/pageweight-yui0114/
À la carte Dependencies

YUI component filesize footprints, including all dependencies — even optional dependencies — for each component. Filesize includes minification and gzipping of all files.
Lightweight CSS Foundation

<table>
<thead>
<tr>
<th></th>
<th>Full, commented file source</th>
<th>Minified file source</th>
<th>Minified, gzipped file source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reset CSS</td>
<td>0.5KB</td>
<td>0.5KB</td>
<td>0.4KB</td>
</tr>
<tr>
<td>Fonts CSS</td>
<td>0.8KB</td>
<td>0.4KB</td>
<td>0.3KB</td>
</tr>
<tr>
<td>Grids CSS</td>
<td>2.5KB</td>
<td>1.8KB</td>
<td>0.6KB</td>
</tr>
</tbody>
</table>
Why Open Source?
What about the timeline?

- **2005, January**
  - Initial development
  - One-person team
- **2005, July**
  - Initial internal release
- **2006, February 13th**
  - Open-source release
  - #1 on Delicious, Digg, Techmeme
  - 36,000 downloads
- **2006, September 18th**
  - 3000th member on [ydn-javascript](http://ydn-javascript)
- **2006, November 7th**
  - Linux Journal Editor’s Choice Award: **Best Software Library**
  - 300k downloads
- **2006, November 13th**
  - Current release
  - 9 person team
What about the namespace?
(because I really like $)
Global Variables are Evil

- Ideally, only a single global per app|lib|widget
- Speed unaffected; self documentation; reliable
- Shorten locally if you want.

- `var $ = YAHOO.util.Dom.get;`

- [http://yuiblog.com/blog/2006/06/01/global-domination/](http://yuiblog.com/blog/2006/06/01/global-domination/)
What’s in the distribution?

• 100s of files
  – All examples
  – All documentation
  – All development builds
  – All production builds
  – All minimized builds
Yahoo! is powered by the exact same bits we offer to you.
What is the license?

The most-open there is: BSD
Who’s Using It?
External Implementations

- Wall Street Journal
- Technorati
- IndyCar.com
- SugarCRM
- SmugMug
- PayPal
- eBay
- Yuriz
- PowerReviews.com
- Madonet
- You?
Maximum Results Displayed

When query results return, an HTML unordered list element (<ul>) of up to ten items (<li> elements) are rendered into the container that you specify. If fewer than ten results are returned, the container’s <ul> element will only have that number of <li> elements. Keep in mind that the height, width, and other style characteristics of the container are determined by CSS. You may use the following code to change the maximum number of <li> elements that may populate the <ul> element:

```javascript
// Display up to 20 results in the container
myAutoComp.maxResultsDisplayed = 20;
```

Minimum Query Length

By default, as soon as a user starts typing characters into the input element, the AutoComplete control starts batching characters for a query to the DataSource. You may increase how many characters the user must type before triggering this batching, which can help reduce load on a server, especially if the first few characters of the input string will not produce meaningful query results. A value of 0 will enable null or empty string queries, which is particularly useful in conjunction with the Always Show Container feature. A negative number value will effectively turn off the widget.

```javascript
// Require user to type at least 3 characters before triggering a query
myAutoComp.minQueryLength = 3;
```

Query Delay

You can also control how long the user must wait before the queries are sent to the server. By default, batching is disabled and you must explicitly call the sendBatch method to send the query. You can control this value directly:

```javascript
// Reduce the delay to 100 milliseconds
myAutoComp.queryDelay = 100;
```
Complete API Documentation

Class `YAHOO.widget.AutoComplete`

The `YAHOO.widget.AutoComplete` class provides the customizable functionality of a plug-and-play DHTML auto completion widget. Some key features:

- Navigate with up/down arrow keys and/or mouse to pick a selection
- The drop down container can "roll down" or "fly out" via configurable animation
- UI look-and-feel customizable through CSS, including container attributes, borders, position, fonts, etc.

Constructor

`YAHOO.widget.AutoComplete( elInput, elContainer, oDataSource, oConfigs )`

Parameters:

- `elInput <HTMElement>`: DOM element reference of an input field.
- `elInput <String>`: String ID of an input field.
- `elContainer <HTMElement>`: DOM element reference of an existing DIV.
- `elContainer <String>`: String ID of an existing DIV.
- `oDataSource <Object>`: Instance of `YAHOO.widget.DataSource` for query/results.
- `oConfigs <Object>`: (optional) Object literal of configuration params.

Properties

- `_aListItems - private Array`
  Array of `<li>` elements references that contain query results within the results container.
- `_bContainerOpen - private Boolean`
  Whether or not the results container is currently open.
Subclassing Panel to Create PhotoBox

Container classes can be subclassed to create all kinds of rich custom controls. The PhotoBox, which we will build in this tutorial, is an example of how Panel can be subclassed and styled to create a basic popup photo viewer with back and forward navigational controls.

The first step to subclassing the Panel is writing the constructor for the new subclass (PhotoBox, in this case) and specifying its inheritance from the Panel class using `YAHOO.extend`:

```javascript
// BEGIN PHOTOBOX SUBCLASS
YAHOO.widget.PhotoBox = function( el, userConfig ) { 
    if (arguments.length > 0) { 
        YAHOO.widget.PhotoBox.superclass.constructor.call(this, el, userConfig);
    }

    // Inherit from YAHOO.widget.Panel
    YAHOO.widget.extend(YAHOO.widget.PhotoBox, YAHOO.widget.Panel);
}
```

Next, we will define a few constants for use by the PhotoBox class: `CSS_PHOTOBOX`, which defines the CSS class to apply to the Panel, and `NAV_FOOTER_HTML`, the HTML that will be used for the footer navigation.

```javascript
// Define the CSS class for the PhotoBox
YAHOO.widget.PhotoBox.CSS_PHOTOBOX = "photoBox";
```
The YUI Cheat Sheets
October 20, 2006 at 7:42 am by Eric Miraglia | In Development |

Update 20 October 2006: This video was originally posted with no link to the slide deck that Douglas uses in the talk. That deck is in PowerPoint format and can be downloaded here.

Douglas Crockford is Yahoo!’s leading JavaScript Architect. He has written extensively on JavaScript and has been among the protagonists of the JavaScript developer community for more than a decade. Douglas is the discoverer of the JSON data format and a frequent contributor to YUIBlog.

At Yahoo!, Douglas has been teaching a series of classes on the JavaScript programming language. These classes bear his unique style and perspective and benefit from the depth of experience he has gained over the years.
Is there a community?

- http://yuiblog.com
- http://groups.yahoo.com/group/ydn-javascript
- http://www.jackslocum.com/yui/ YUI.ext
- http://blog.davglass.com/ YUI.addons
- SourceForge
When I first started looking at the Yahoo! UI library, one of the things I noticed was a lack of real world examples. Sure, the examples make little boxes fly and fade, make them drag and drop and other cool effects but how do you take that and put it into your website or application? This blog is going to be dedicated to demonstrating how to do that.

While writing this blog a small library of reusable components based on YUI has been started. The library was named spaced YAHOO.ext (short for Yahoo! UI extensions), and already has a few very useful classes that make day to day development with YUI much easier. All of the classes are well-documented and there are some examples of using them found in the posts on this blog. The code is of course free and has the same unrestricted (BSD) license as Yahoo! UI.

Jack

"Build your own" Download Dialog and RC1 Release
November 11, 2006 by Jack Slocum | 33 Comments »

This post is to notify everyone the .33 RC1 build is up. This release is primarily bug fixes. There's also a new download dialog with a "Build your own" feature that I think demonstrates how easy it is to rapidly build something with YUI + yui-ext. The download dialog consists of a modal BasicDialog with a BorderLayout embedded. It uses a YAHOO.ext.UpdateManager with a JSON renderer to build the list of options. It's … more »
How is it different from GWT?
We believe in JavaScript.
Thanks!
nate.koechley.com/talks/2006/11/refresh06/YUI/

• Nate Koechley:
  – nate@koechley.com | natek@yahoo-inc.com
  – http://nate.koechley.com/blog

• Yahoo! Developer Network & Y! UI Blog:
  – http://developer.yahoo.com
  – http://developer.yahoo.com/yui
  – http://developer.yahoo.com/ypatterns
  – http://www.yuiblog.com
  – http://groups.yahoo.com/group/ydn-javascript
We’re Hiring!

Josie Aguada: jaguada@yahoo-inc.com

Usual suspects:

JavaScript, PHP, CSS, HTML, ActionScript…