# YUI 3.x Architecture (JavaScript)

based on YUI 3.0 PR1 2008-09-12

developer.yahoo.com/yui

## seed

### YUI

**YUI Base**
Minimal YUI Global and core library methods. The only universal YUI 3.x dependency.

**Get**
Provides generic mechanism for dynamically loading script and CSS resources.

**Loader**
Adds dependency calculator and YUI 3.x package metadata, allowing dynamic loading of all YUI modules in single, combo-handled HTTP requests.

## core

### Event

**Event Base**
Provides DOM event and Custom Event functionality, event facade implementation, and page-load storyboarding events.

### Dom

**Dom Base**
Provides low-level DOM helper methods.

**Dom Style**
Provides low-level style management methods.

**Dom Screen**
Provides low-level positioning and region management methods.

**Selector**
Provides low-level methods for collecting and filtering DOM elements based on selector syntax.

### OOP

**OOP Base**
Provides low-level object management functionality for JavaScript objects including extension and augmentation.

### Node

**Node Base**
Provides abstraction for creating and manipulating DOM elements. Can be extended by components via plugin architecture.

**NodeList**
Extension of Node to handle abstraction for element collections; used for batching DOM manipulations.

**Node Style**
Extended API for manipulating style properties on Nodes and NodeLists.

**Node Screen**
Extends Node and NodeList API for positioning and region management.

## component framework

### Widget
Provides foundational plumbing for high-level UI components, including lifecycle management. Builds upon base and is an Attribute provider.

### Base
Provides base class for all YUI components that provide managed attributes and that serve as event targets.

### Attribute
Provides managed attribute system that can augment any class; includes get/set interface and intrinsic change events.

### Plugin
Plugin namespace. Forthcoming class will serve as a template for plugin functionality. Plugins can also operate directly on Nodes.

## components

### Drag and Drop

**DD Manager Base**
Provides base DD manager functionality to make a Node draggable.

**DDM Shim**
Extends DD manager to allow elements to be dragged over iframes and other mousetraps.

**DD Drag**
Provides DD functionality for simple drag operations on a Node.

**DD Drop**
Provides drop-target functionality.

**DD Proxy**
Provides proxy-drag functionality.

**DD Constrain**
Provides drag constraint functionality.

**DD Plugin**
Provides pluggable drag functionality for Node instances.

**DD Drop Plugin**
Provides pluggable drop-target functionality for Node instances.

### Animation

**Anim Base**
Provides foundational animation support for animating properties on the style object of an element.

**Anim Color**
Extends animation functionality to support animations of color values.

**Anim Curve**
Extends animation functionality to support animations of element position along a curved path.

**Anim Easing**
Extends animation functionality to support easing effects.

**Anim Scroll**
Extends animation functionality to support animations of elements' scroll positions.

**Anim XY**
Adds support for the xy property in from and to configurations.

**Anim Plugin**
Provides pluggable animation support for Nodes.

### IO
Provides XMLHttpRequest, SWF-based cross-domain requests, and file upload support.

### Cookie
Provides cookie management support.

### JSON

**Stringify**
Convert JavaScript objects to JSON strings.

**Parse**
Safely convert strings to JavaScript objects.

### ...and more
More utility (low-level) components are provided in the YUI 3.0 preview release 1 build. Future preview releases and betas will include the full component framework and samples of higher-level components (widgets).

---

*The YUI Seed layer can be used to bootstrap any implementation. With the seed on the page, YUI().use() can bring in all additional components quickly and safely.*

*YUI's 3.x Core serves as a common dependency layer for most downstream components. Let the YUI Loader (or Dependency Configurator) calculate your true core dependencies, though; in many cases, you will not require the full core layer, even if you are using downstream components.*

*The Component Framework (not yet complete as of PR1) will provide a consistent and reusable structure for building low-level utilities and higher-level UI widgets.*

*Components in YUI 3.x are provided a la carte and can be highly modular. You can minimize your inclusion of library code by selecting the specific portions of each component that are required in your implementation.*