



# Rich Client Interfaces: *Breaking Free from the Browser*

*Kevin P. Wojdak, Senior User Interface Developer*

## Breaking Free from the Browser

Pathfinder Associates creates rich client interfaces in Internet browser windows that break applications free from the constraints of a browser window. Using tried and true methods of rich client interface development, Pathfinder builds application interfaces that work in browser windows but don't look like browser-based interfaces. We build application interfaces that look like applications.

Through years of practical development and the constant desire to be unique, Pathfinder has constructed many creative and interesting interface objects that defy the basic functionality provided by browsers. By stretching the imagination and the limitations of the browser, we have found ways to go beyond simple browser capabilities to provide functionality that rivals application functionality built by traditional development tools.

For example, Pathfinder has built the following types of objects and application functionality for projects:

### *Search and Sort field*

The Search and Sort field object is a combination of input fields and code that allow a user to enter a single or several characters into a field and, based on a predefined list of data, receive back a list of every item from that list that contains the entered text. Our Search and Sort field object goes one step further by sorting the returned list alphabetically so that the items where the search characters appear nearer to the beginning of the list item, appear higher on the list and stair step down through the results list until you get the items where the search characters appear at the end of an item. This field appears as a simulated dropdown list on screen.

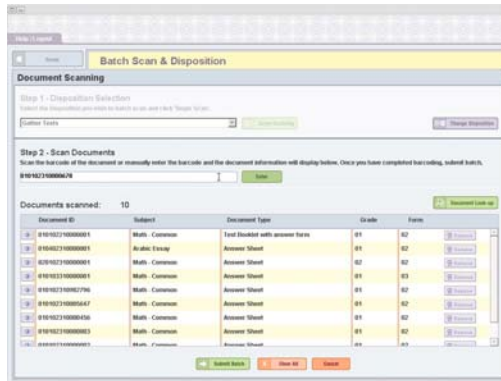


The screenshot displays a simulated dropdown list interface. At the top, there is an input field containing the text 'blu' and a cursor. To the right of the input field is a button labeled 'Select Text'. Below the input field, a list of search results is displayed, each line starting with a bolded letter 'B' and containing text that includes the search term 'blu'. The results are:

- Blue** is the color I get when sad
- The blue** boy went that way
- I like** the blue truck
- John had** the blues over mary
- Mary Mary** was a blue fairy
- You have** beautiful blue eyes
- Show me** a good time blue man group
- I love** to listen to the Blues
- When I get** back from the blue man group show
- Quit lolly-gagging** around this bluebird tree

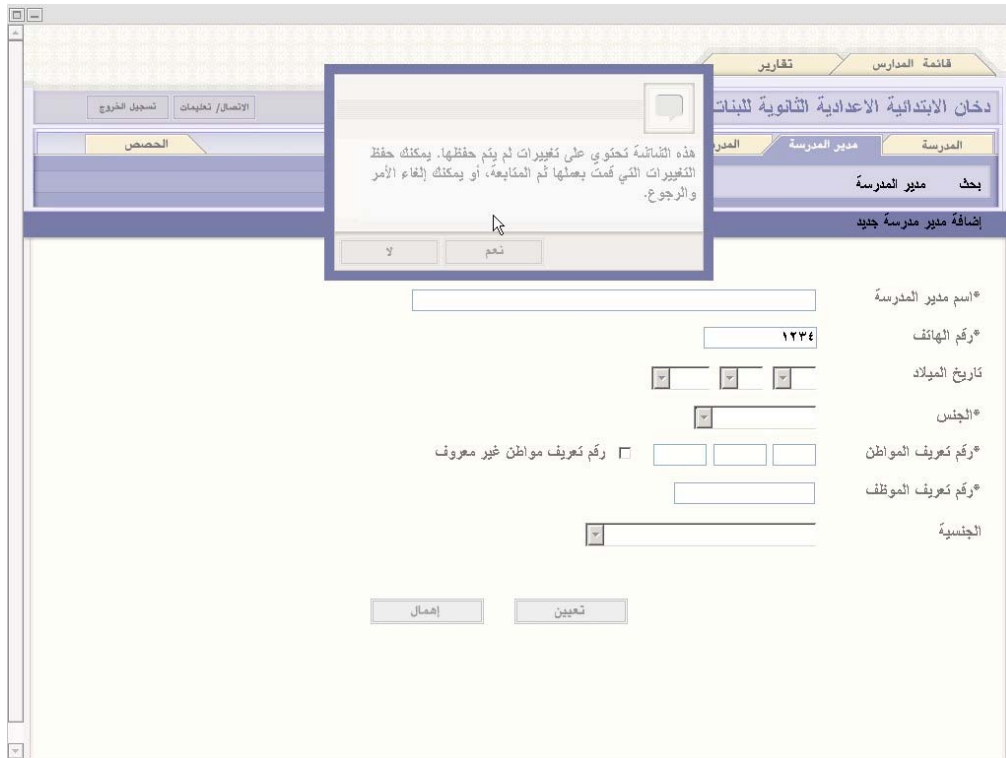
**Interchangeable Localized Interfaces (One set of code to manage)**

Using the same single interface page and source code, Pathfinder can change the look and feel of the page based on the current language being displayed. Pathfinder detects the current language being used and supplies the user with an appropriate interface based on their language needs utilizing a single code base. i.e., The Arabic language reads right-to-left whereas the English language reads left-to-right. Pathfinder builds interfaces that will do both at the flip of a switch using the same page of code. For the Arabic interface, all fields shift to the right so that the fields appear to read from the right, type from right-to-left, type using an Arabic font, and the scrollbar appears on the left.



### DHTML Modal Dialogs

Windows functionality provides a user with the ability to display messages in a modal dialog but, that dialog looks and acts like a windows modal dialog object (modal dialog boxes are dialog objects used to display information that has to be acted upon or the dialog closed before any other action can take place). Pathfinder can build a fully customized DHTML dialog box that acts like a windows modal dialog without actually using Windows modal dialog functionality.



### Dynamically Growing Tables without Page Refresh

At the click of a button or simply through the act of entering data, we can dynamically expand a table of data on screen without refreshing the page. A good use for this functionality is to gather a group of data on screen and then submit this data as a batch to a backend process. Why acquire just one address in a form and then submit it to the server when you can get all addresses entered in one dynamically growing list and submit the group?

If yes, please fill out the following.

Date of Loss	Employee First Name	Employee Last Name	Department
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<a href="#">Remove Row</a>			Total Amount of Incurred Loss (\$US 000)
Location	Description of Occupational Disease	Cause of Occupational Disease	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Date of Loss	Employee First Name	Employee Last Name	Department
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<a href="#">Remove Row</a>			Total Amount of Incurred Loss (\$US 000)
Location	Description of Occupational Disease	Cause of Occupational Disease	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Date of Loss	Employee First Name	Employee Last Name	Department
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<a href="#">Remove Row</a>			Total Amount of Incurred Loss (\$US 000)
Location	Description of Occupational Disease	Cause of Occupational Disease	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

[INSERT NEW LINE](#)

#### 15 (of 33) Product Recall

Have any of your manufactured products ever been recalled? (Do not include products that are manufactured by another company that you just distribute.)

Yes      No

## Barcode Scanner Interface

Using HTML and JavaScript, we created a barcode scanner interface that accepted an input stream from a handheld barcode scanner, validated the stream, broke it down into component pieces, and then displayed the data on screen in a dynamically growing table. After all codes were gathered from documents in a batch, the entire list of barcodes was submitted to the server for processing. This process benchmarked at 35-40 barcodes per minute or 2100-2400 documents scanned per hour.

The screenshot displays a web application interface for document scanning. At the top, there is a navigation bar with 'Help | Logout' and a 'Home' button. The main heading is 'Batch Scan & Disposition'. Below this, the interface is divided into two steps:

**Step 1 - Disposition Selection**  
Select the Disposition you wish to batch scan and click 'Begin Scan'.  
A dropdown menu shows 'Gather Tests'. To its right is a green 'Begin Scanning' button and a 'Change Disposition' button.

**Step 2 - Scan Documents**  
Scan the barcode of the document or manually enter the barcode and the document information will display below. Once you have completed barcoding, submit batch.  
An input field contains the barcode '010102310000678' and an 'Enter' button.

Below the input field, it says 'Documents scanned: 10' with a 'Document Look-up' button.

Document ID	Subject	Document Type	Grade	Form	
010102210000001	Math - Common	Test Booklet with answer form	01	02	Remove
010402310000001	Arabic Essay	Answer Sheet	01	02	Remove
020102310000001	Math - Common	Answer Sheet	02	02	Remove
010103310000001	Math - Common	Answer Sheet	01	03	Remove
010102310982796	Math - Common	Answer Sheet	01	02	Remove
010102310005647	Math - Common	Answer Sheet	01	02	Remove
010102310000456	Math - Common	Answer Sheet	01	02	Remove
010102310000003	Math - Common	Answer Sheet	01	02	Remove
010102310000002	Math - Common	Answer Sheet	01	02	Remove

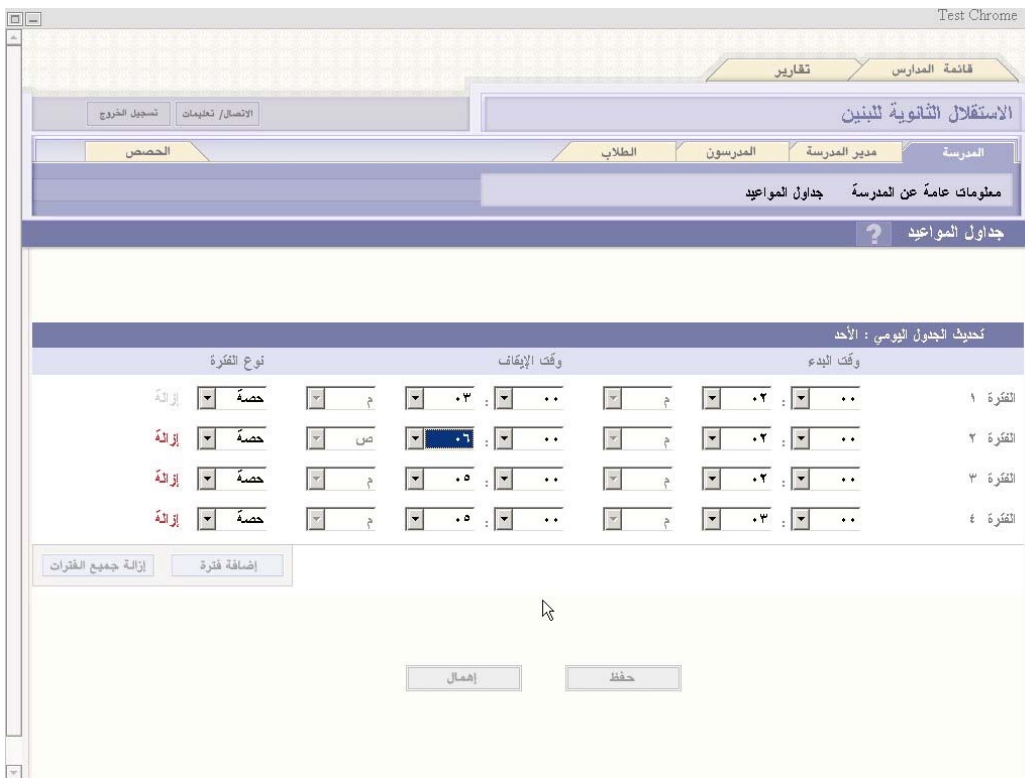
At the bottom of the table, there are three buttons: 'Submit Batch' (green), 'Clear All' (orange), and 'Cancel' (orange).

### Error Sounds

For the barcode scanner, Pathfinder had to devise a way to respond to the user when a bad or duplicate barcode was scanned. For this task, two different sounds were selected and as the barcodes were validated, either a single error sound was heard for a bad barcode or a double beep sound was heard for a duplicate code.

### Chromeless Window

One requirement for an application was to mask the fact that the application was running through a Microsoft Windows Internet Explorer browser window. The requirement was for the entire browser-based standalone client/server application to appear as an application and not as a website in a browser. One reason for this was the need for security by not allowing a user to access the source code as can be done with a browser window. Pathfinder's solution was to present the application in a fully customized Chromeless window. A Chromeless window is an interface framework that looks like an application without having Microsoft standard information displayed in the title bar, all colors customized, no borders or status bar, etc. The idea is that if an interface doesn't look like it's in a browser window or interacts like a web page, then it gives the impression of the interface being a true interface for a compiled application.



### ***Drilldown List with Filtering Capabilities***

One very technical requirement involved the creation of a single standalone application page that would display a list of all students in a particular school that could be drilled down to the grade and then to a vocational track and section within each grade level and finally to the individual student records. Each student was then linked to his or her student information record. Additionally, this list had to allow a filtering capability to show only a single grade, track, and/or section. Pathfinder achieved this complex page using several techniques:

#### **Folding Table**

To create the effect of a drilldown list, the students were added to a large HTML table that was then manipulated in such a way as to appear folded. Students within a section were hidden when viewing just a section and sections were hidden when just grades or tracks were visible, etc.

#### **One-to-Many-to-Many Data Relationships on the Client Side**

For the filtering task, a method of relating the data had to be devised that could then be acted upon. The data had to be related so that an individual student could be matched with an individual vocational track and then section within a grade. For each grade, there were one or more tracks each with multiple sections and containing multiple students. The sections while different under each track, shared a common naming convention. To create this complex relationship, Pathfinder defined a mini-database within the client code populated when the page was loaded. The database was defined using a unique multi-dimensional data structure which when populated resided in memory for easy access. Three dynamic dropdown list fields controlled the filtering. By selecting a grade from the first dropdown, the vocational track dropdown was populated. Based on the track selected, a unique list of sections would populate the section list. After each selection, the table would fold and unfold to reveal only the requested filtered data.

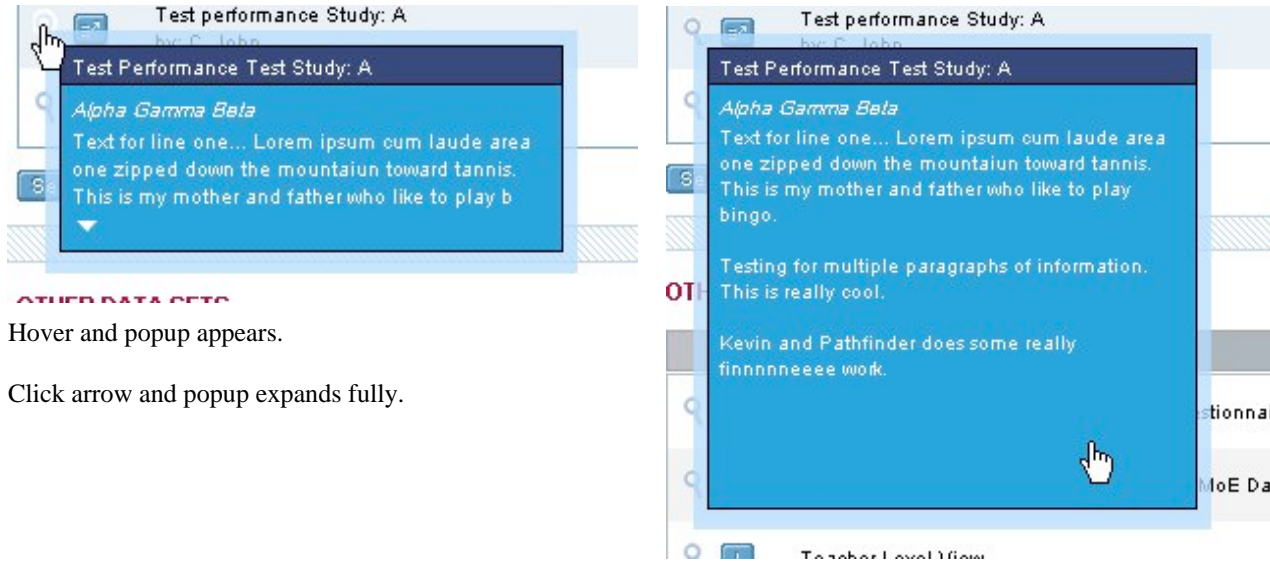


Drilldown List Example:

قائمة المدارس		تقارير	
دخان الابتدائية الإعدادية الثانوية للبنات			
المدارس		المحاص	
قائمة التلاميذ		بحث	
قائمة التلاميذ			
إضافة تلميذ جديد			
تصفية قائمة التلاميذ حسب			
كل الصف	و/ أو	كل التخصص	و/ أو
الصف	التخصص	الشعبية	
1 ▼			
كل الشعبية ▼			
الاسم	رقم تعريف التلميذ	رقم تعريف المواطن	الجنسية
0 ◀			
1 ◀			
2 ◀			
3 ◀			
4 ▼			
كل الشعبية ▼			
الاسم	رقم تعريف التلميذ	رقم تعريف المواطن	الجنسية
1 ▼	00029402384	29463405351	قطر
كسليم محمد صالح كحلته المرعي			
نقي حافظ غياث الدين شودي	00029401818	2940500017	بنجلادش
حبيبة صيفر سالم ابومكريعه المنصوري	00029402353	29463404710	قطر
داله خلفان علي راشد النعيمي	00029402386	29463402498	قطر
رحمة راشد عبيد راشد السناري	00029500235	29563400575	قطر
روزه راشد علي السندي النعيمي	00029402360	29463402432	قطر

### Expandable Information Popup

For more information within a portal application, a user could place a mouse pointer over an icon. After 3 seconds, a small window would popup showing more information. But, this small window only showed so much information. An arrow at the bottom of the popup allowed the popup to be expanded further. This effect utilized timers, reusable popup with data loaded externally, coordinate mapping to display the popup exactly where the mouse pointed, page size awareness to pop down or up depending on the location of the mouse on the screen, and the ability to increase the size of the popup while it hovered on top of the page. Once the mouse moved off the popup, it disappeared from the screen.



Hover and popup appears.

Click arrow and popup expands fully.

## Floating Moveable Post-It Type Information Popup

Using only code and markup, a mouse pointer hovers over an icon and an information note pops up to give the user directions on how to use the page. Moving the mouse off the note closes it. By clicking on the note, it becomes sticky on the page and can be moved around and set aside to provide direction while the user adds data to the page.

The screenshot displays a web application interface with a floating information popup. The interface includes a navigation bar with tabs for 'تقارير' (Reports) and 'قائمة المدارس' (School List). Below this, there are sections for 'دخان الابتدائية الإعدادية الثانوية للبنات' (Dhahran Secondary School for Girls) and 'الحصص' (Subjects). The main content area is titled 'السجل المُعين للتلميذ في هذه المدرسة' (The designated record for the student in this school) and contains a 'جدول المواعيد' (Schedule) section. The schedule section includes a list of instructions for adding and managing events in the calendar. At the bottom, there are three text input fields for 'ملاحظات عن التلميذ / العائلة' (Notes about the student/family), 'ملاحظات أخرى' (Other notes), and 'ملاحظات على جدول التلميذ' (Notes on the student's schedule). There are also buttons for 'إغلاق' (Close), 'إرسال' (Send), 'حفظ' (Save), and 'إلغاء تعيين التلميذ' (Unassign student).

تقارير قائمة المدارس

دخان الابتدائية الإعدادية الثانوية للبنات

الطلاب المدرسون مدير المدرسة المدرسة

الحصص

الوحدة للتقرير

معايير البحث السابق قائمة التلاميذ بحث

تحديث تلميذ آخر

السجل المُعين للتلميذ في هذه المدرسة

جدول المواعيد

تتمثل أول خطوة لإنشاء جدول مواعيد في إضافة فترات مسكّنة إلى جدول المواعيد المقدم من المدرسة. سنقوم في هذه الشاشة بما يلي:

1. أدخل زمن البدء
2. أدخل زمن الانتهاء
3. أدخل نوع الفترة لكل فترة.

لإضافة أكثر من فترة واحدة، عندما يكون هناك فترة بالفعل في جدول المواعيد، انقر فوق زر "إضافة فترات".

لإزالة فترة واحدة على حدة، انقر فوق زر "إزالة" المقابل لكل فترة.

لإزالة جميع الفترات مرة واحدة، انقر فوق زر "إزالة جميع الفترات".

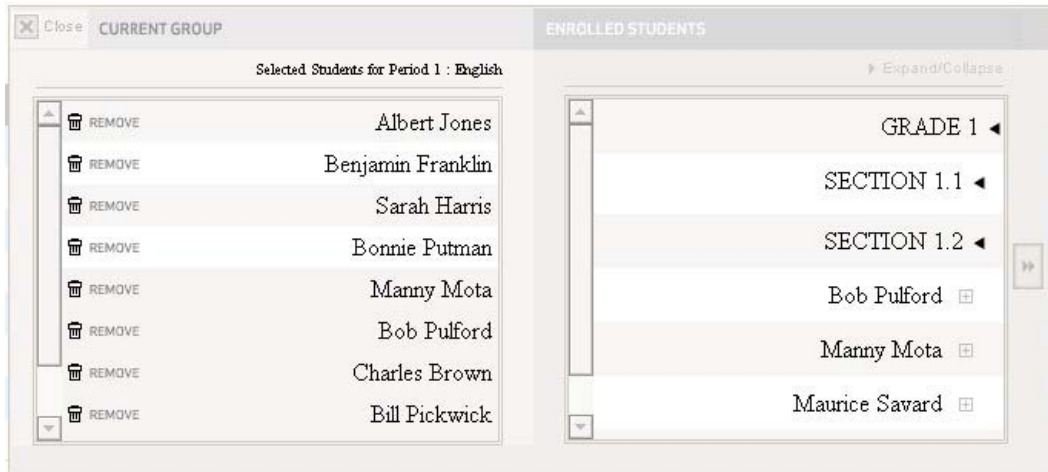
4. بعد استكمال إضافة كافة الفترات التي تريدها، انقر فوق "إنشاء جداول مواعيد" وسيقوم التطبيق بإنشاء جداول مواعيد متكاملة لكل يوم.

ملاحظات عن التلميذ / العائلة ملاحظات أخرى ملاحظات على جدول التلميذ

إغلاق إرسال حفظ إلغاء تعيين التلميذ

### Dragging and Dropping Data

To allow groups of data to be moved from one area of a window to the next in as few keystrokes as possible, Pathfinder implemented a drag and drop function for a project. We accomplished this task using only basic HTML and client-side scripting on a popup window. This data for a school drilled down to grades, class sections within those grades, and then to individual students. The drag and drop capabilities allowed for individual students, sections, or even whole grades to be dragged to the other area.



### Pathfinder Interface Development Style

Pathfinder builds interfaces using common freely accessible tools and languages that anyone can use. There is no need to purchase expensive development suites to achieve the creative results Pathfinder provides to our clients.

Our developers think on both a creative level as well as a technical level. To achieve the best results for our clients, our designers are limited only by their imaginations and our developers transform those imaginative designs into technical reality through technical creativity. We are only limited by how creative we can be technically. Creative solutions require technical creativity to achieve positive results.

Elements on a creative interface are broken down to component pieces using the technical idea of an object. Each element becomes an object that is manipulated on its own and contains embedded behaviors and actions. By taking an object-oriented approach to interface development, Pathfinder is able to build solid cross-platform interfaces while still providing living reality to the imaginative ideas of our designers.

We combine creative source code development with world-class interface design and creativity to produce unique and user-friendly interfaces.