

# Ten XPages Design Patterns

Matt White - Elguji Software



# About Me

Independent Notes, Domino and Java developer from London

Been doing Lotus development since 1996, independent since 2000

Lead Developer for Elguji Software the makers of IdeaJam and IQJam

Recently launched XPages Training site: [XPages101.net](http://XPages101.net)

All over the place online:

- Blog: [mattwhite.me](http://mattwhite.me)
- Twitter: [@mattwhite](https://twitter.com/mattwhite)
- Skype: [whitemrx](https://www.skype.com/en/contacts/skype/whitemrx)
- Email: [matt.white@elguji.com](mailto:matt.white@elguji.com)



# What is a design pattern?

## Design pattern (computer science)

From Wikipedia, the free encyclopedia

In [software engineering](#), a **design pattern** is a general reusable solution to a commonly occurring problem in [software design](#). A design pattern is not a finished design that can be transformed directly into [code](#). It is a description or template for how to solve a problem that can be used in many different situations. [Object-oriented](#) design patterns typically show relationships and [interactions](#) between [classes](#) or [objects](#), without specifying the final application classes or objects that are involved.

Not all software patterns are design patterns. [Algorithms](#) are not thought of as design patterns, since they solve [computational](#) problems rather than [software design](#) problems. [Architectural patterns](#) are larger in scope, usually describing an overall pattern followed by an entire program. [Programming paradigms](#) describe a style which can be the basis for an entire programming language.

## So in our terms?

## We're looking for ways to do frequently required tasks such as...



# Agenda

## **Some Simple Examples**

- Linking Combo Boxes
- Create the equivalent on an @Picklist
- Pass complex data to the browser to use in Client Side Javascript

## **The geeky stuff**

- Some advanced validation
- Generating non HTML XPages

## **Taking things a little further**

- AJAX Get and Post using Dojo
- AJAX Login Box
- Securing an XPage
- Building Complicated Tables
- Logging Errors

## **Bonus Tips**

- Hooking onto data events
- XPages and IE8 in 8.5.0





# Don't panic

No need to copy down the code, all of it's available as a download:

[http://mattwhite.me/blug\\_tenpatterns.zip](http://mattwhite.me/blug_tenpatterns.zip)



# Linking Combo Boxes

## What we're trying to do

- Two combo boxes
- Options in the second combo box are dependent on the value selected in the first one

## How to do it

- Add your combo boxes to the XPage
- Build your values lists
- Go to the first combo box and make it a partial refresh with no data validation
- Change the computed values list in the second box to be dynamically calculated rather than static (# vs \$)



# Linking Combo Boxes (cont.)

Region: Please Select a Region ▼

Country: { Computed Items } ▼

Events

- Mouse
  - onclick
  - onmousedown
  - onmouseup
  - onmouseover
  - onmousemove
  - onmouseout
- Focus
- Select & Change
  - onchange**
- Key

Client Server

Simple Actions Script Editor

Add Action... Add Group... Edit... Remove

Action	Arguments
Simple Action	

Server Options

- Full Update
- Partial Update**
- No Update
- No Submission

IDs: country

☒ No data validation

Edit Event Parameters...

```
try{
    var region = getComponent("region").getSubmittedValue();
    var list = @DbLookup(@DbName(), "vwRefData", region, 2);
    if (@IsError(list))
        return "Please select a region first|";
    else
        return list;
}catch (e){
    return "Please select a region first (Error)|";
}
```

```
<xp:comboBox id="country">
    <xp:selectItems>
        <xp:this.value><![CDATA[{#{v} vascript:try{
            var region = getComponent("region").getSubmittedValue();
            var list = @DbLookup(@DbName(), "vwRefData", region, 2);
            if (@IsError(list))
                return "Please select a region first|";
            else
                return list;
        }catch (e){
            return "Please select a region first (Error)|";
        }}}]></xp:this.value>
    </xp:selectItems>
</xp:comboBox>
```

# Demo





# @Picklist in XPages

## What we're trying to do

- Mimic the @Picklist function that we use in Notes client to display a view and select a value from it

## How to do it

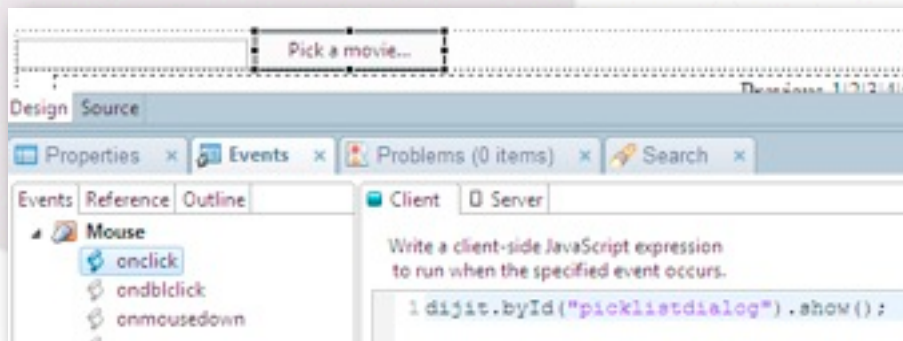
- A little bit involved, we need to use the Dojo Dialog and handle some “features” of XPages
- First add a view with a hidden div around it
- Then add some “onLoad” javascript to manually create the dojo.Dialog widget.
- We can't do it automatically as the XPages AJAX elements will stop working
- Make use of the view “getSelectedIds” method when we click the OK button
- Do a partial refresh of the field we're storing the selected value in

# @Picklist in XPages (cont.)

We have to manually create our Dojo Dialog so that XPages Javascript continue to work.

So we reference our div called “picklistdialog” in an “addOnLoad” function:

```
1 /**
2  * Creates a dijit dialog box based on a div content
3  * and moves the dialog back inside the form so that partial
4  * refreshes can work properly
5  * @param id div identifier
6  */
7 function dialog_create(id) {
8     var dialogWidget = dijit.byId(id);
9     if( dialogWidget )
10         dialogWidget.destroyRecursive(true);
11     dialogWidget = new dijit.Dialog(
12         { },
13         dojo.byId(id)
14     );
15     var dialog = dojo.byId(id);
16     dialog.parentNode.removeChild(dialog);
17     var form = document.forms[0];
18     form.appendChild(dialog);
19
20     dialogWidget.startup();
21 }
22
23 dojo.addOnLoad(function() {
24     dialog_create("picklistdialog");
25 });
```

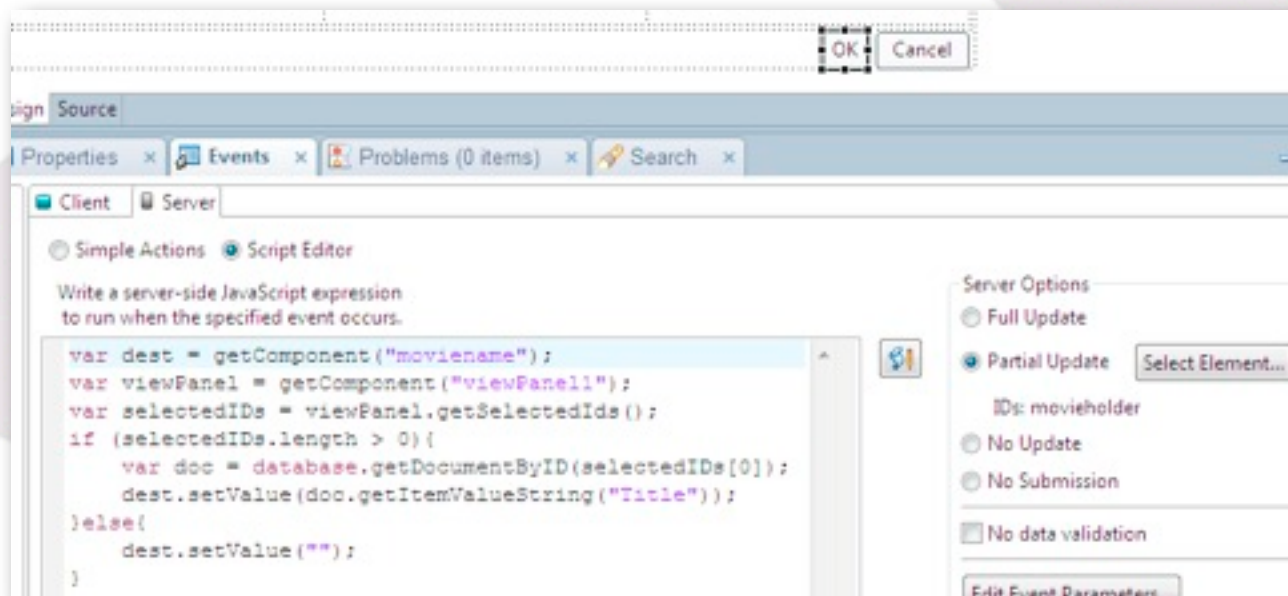


Then when the Pick a Movie button is clicked we can just “show” the dialog which has been built.

# @Picklist in XPages (cont.)

In the dialog, when the OK button is pressed we perform some Server Side Javascript to get the selected Note ID, get the associated document and store the movie title into the field back in the “main” area of the form.

Then finally we do a partial update of the page to show the newly entered value



# Demo





# Complex Data

## What we're going to do

- Get data out to the browser client so that we can use it in client side javascript (CSJS) functions
- The equivalent of computed text in “classic” web development

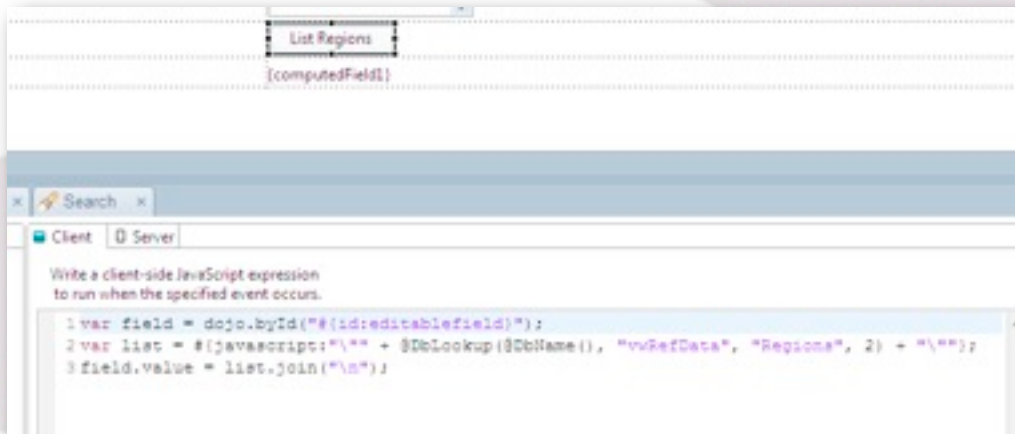
## How to do it

- We're going to use embedded server side Javascript (SSJS) again:  
#{id:myfieldname} and getClientId(“myfieldname”)  
#{javascript:someServerCode()}
- This can sit in any client side event of a control, or in a computed text object value



# Complex Data (cont.)

In CSJS we can insert server side code using the `# {javascript:myFunction()}` syntax



We can build our own CSJS in a SSJS computed field:



# Demo



# AJAX Get and Post

## What we're trying to do

- We've all used AJAX in other frameworks or rolled our own, how do we do it in XPages
- Dojo is the framework in XPages (1.1.1 in 8.5.0 and 1.3.1 in 8.5.1)

## How to do it

- This is pure Client Side Javascript (CSJS), no real XPages involved.
- The Dojo functions that we're going to use are `dojo.xhrGet` and `dojo.xhrPost`

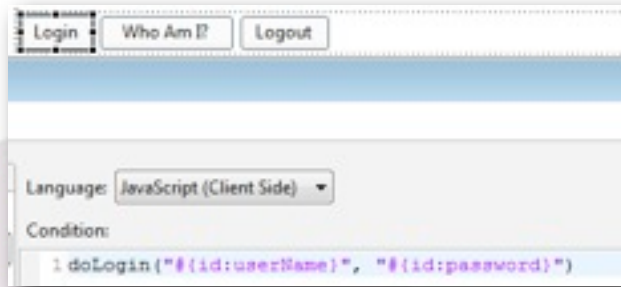
```
1 function getUser(){
2     dojo.xhrGet({
3         url: dbPath + "/userInfo?open&rnd=" + Math.random(),
4         load: function(data) {
5             dojo.byId("results").innerHTML = data;
6         },
7         error: function(data) {
8             dojo.byId("results").innerHTML = "There was an error: " + data;
9         }
10    });
11 }
```



**BLUG**  
Belux Lotus User Group



# AJAX Login



First add a button to your XPage and add some client side Javascript. Pass the field IDs that we're interested in.

The doLogin function performs a Post AJAX request and then updates the current username

You can do whatever you like after the successful request

NB: This solution does require Session Authentication to be enabled

```
13 function doLogin(userNameId, passwordId){
14     dojo.xhrPost({
15         url: "/names.nsf?login",
16         content: {
17             username: dojo.byId(userNameId).value,
18             password: dojo.byId(passwordId).value
19         },
20         load: function(data) {
21             if (data.indexOf("<input name=\"Username\"") == -1)
22                 getUser();
23             else
24                 dojo.byId("results").innerHTML = "Incorrect Username or Password";
25         },
26         error: function(data) {
27             dojo.byId("results").innerHTML = "There was an error: " + data;
28         }
29     });
30     return false;
31 }
```

# Demo



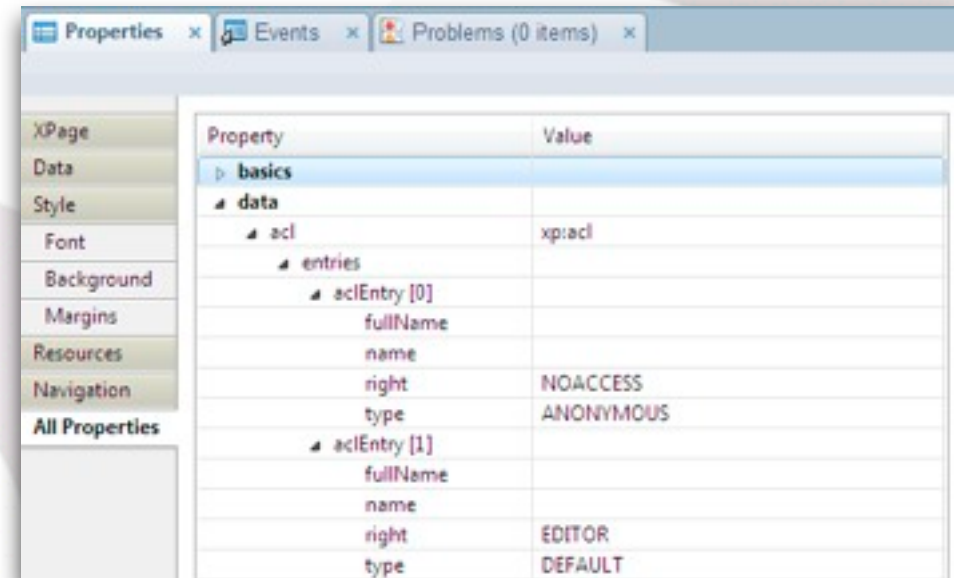
# Securing an XPage

## What we're trying to do

- Prevent non authenticated users access to an XPage
- Restrict an XPage to only certain types of user (by role or group)

## How to do it

- In the XPage “All Properties” go to the data section and define ACL entries for different types of user.
- Can choose between No Access, Reader and Editor rights
- And we can use either user names, groups or roles to identify people.



# Repeating Repeats

## What we're trying to do

- Build a complex HTML table using various sets of data from our Notes database
- An example is the Agenda screen from the UKLUG and ILUG conferences where we have three nested repeat controls!

## How to do it

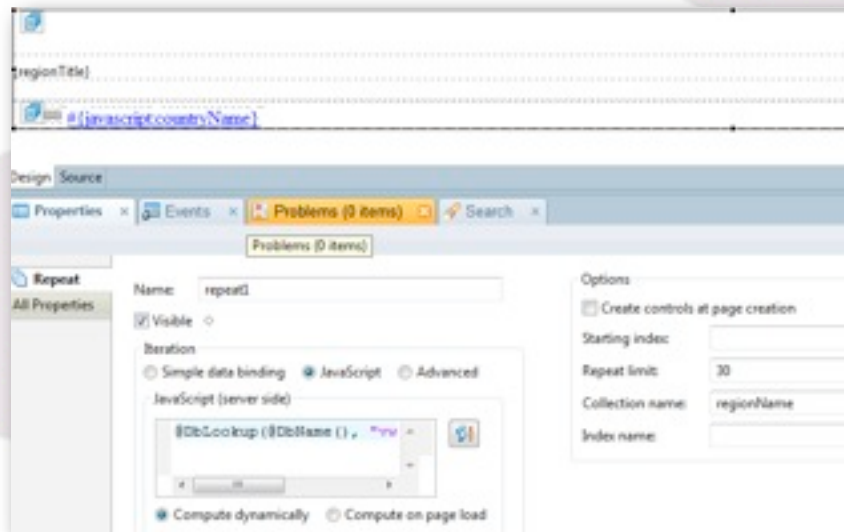
The key to remember is that a repeat control can be used to loop through any list, not just a view, so for example:

- Document Collections
- Arrays (e.g. the result of an @DbLookup)
- Vectors (e.g. a multi value field displayed in a view column)

We just need to make sure we name our variables sensibly, and build correct HTML as we go.

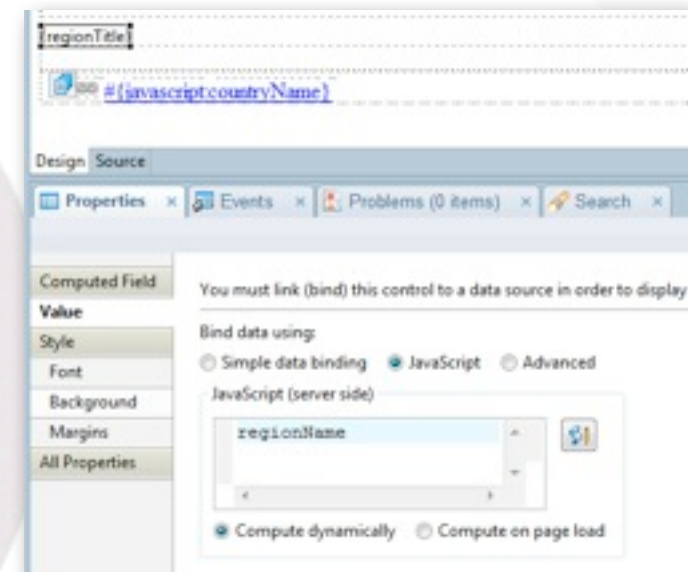


# Repeating Repeats (cont.)



The first repeat does an @DbLookup  
The Collection Name is actually the  
name of the element that we can  
refer to inside the repeat

So for example we can add a  
Computed Field control and reference  
“regionName” and for each iteration  
over the list we’ll get the region name

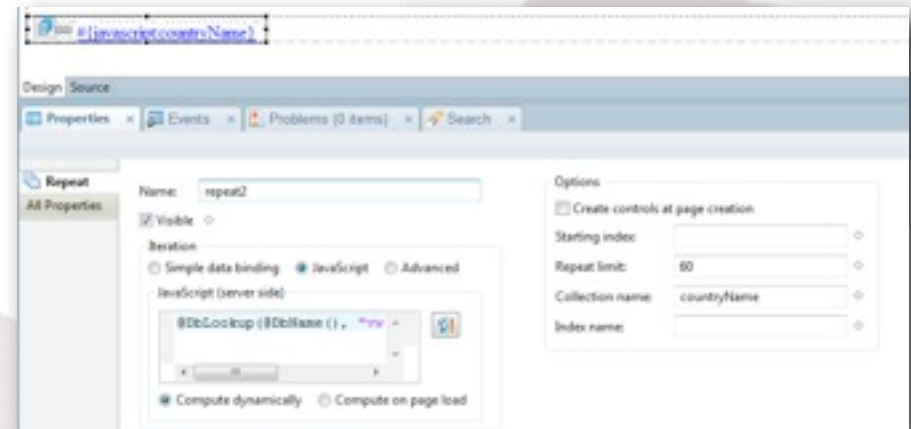


# Repeating Repeats (cont.)

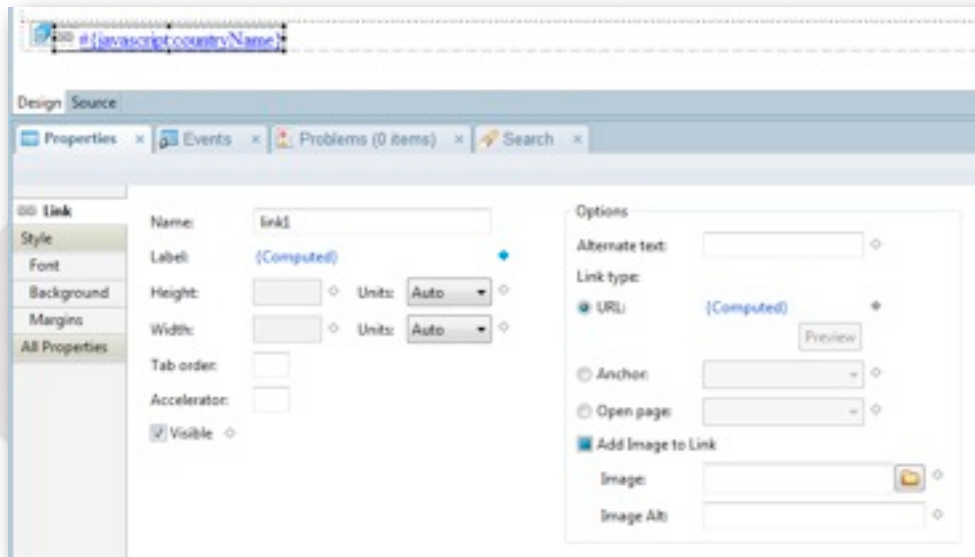
The second repeat can then use the regionName data to build it's own list of data.

We do another @DbLookup to get the country list for each region

Again we create the Collection Name so that we can refer to the values inside the repeat



And then finally inside the second repeat, we create a link for each countryName element returned by the @DbLookup



# Demo



# Error Logging

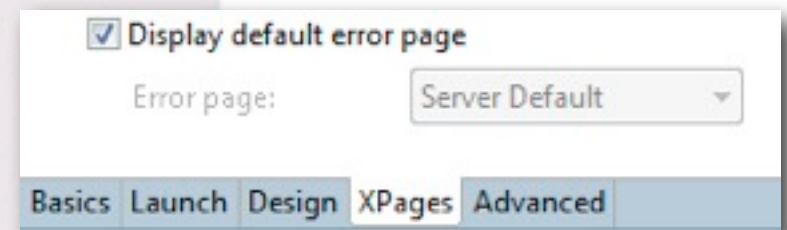
## What we're trying to do

Several things:

- Debug our development process
- Provide useful error messages to the user
- Log useful error messages when running in production

## How to do it

- At the most basic level we have “print”
- And don't forget about the “Display Default Error Page” setting in application properties:
  - This will show errors on the browser
  - (only use during development)
- Also check out OpenLog and Medusa





# Error Logging (cont.)

- To log details of errors or other details of the application you can (should) be using OpenLog
- We have written an XPages version of the OpenLog script library.
- Just include the OpenLogXPages script library into your XPage
- or use “import OpenLogXPages in your own script libraries

```
try {  
    var test = new MySillyCode();  
    test.forceError();  
}catch (e){  
    log.logError(e.getMessage(), SEVERITY_HIGH, e.getMessage(), "", "form_caughterror", "computedField1");  
    return "There has been an error: " + e.getMessage();  
}
```

Logged Event				
Event	Error	User Name	Anonymous	Client Version
Type		Effective User	Anonymous	Release 8.5.0 December 26, 2008
Event Time	24/06/2009 17:40:58	Name		
Start Time	24/06/2009 17:40:58	Access Level		
Severity	2	Database		
		Roles		
Error Num	Error Msg	Script Interpreter error, line=2, col=24: Reference error: 'MySillyCode' not found	Server	lbaasdev01
Error Line	Language		Database	presentation/temp/term...
Back Trace			Agent	form_caughterror
			Method	computedField1
Script Interpreter error, line=2, col=24: Reference error: 'MySillyCode' not found				
Creation Date = 24/06/2009 17:40:57				
Created By = Anonymous				
NotificationSent =				



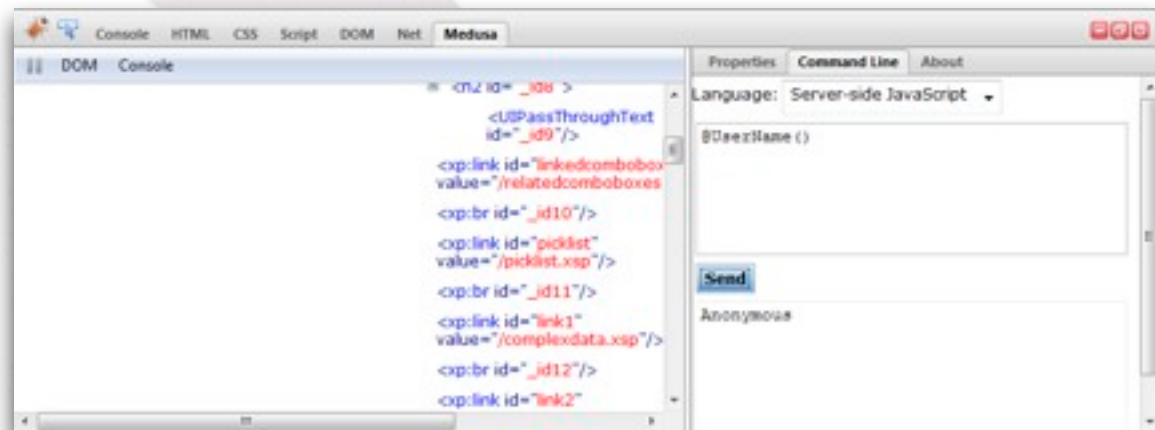
**BLUG**  
Belux Lotus User Group

# Error Logging (cont.)

Medusa is a new OpenNTF project by Tim Tripcony & Nathan Freeman from Group Experts

- Offers integration to Firebug in the browser
- Allows better “live” logging of activity and also offers a command line tool
- Very cool extension to XPages that shows where we’re heading with Domino development

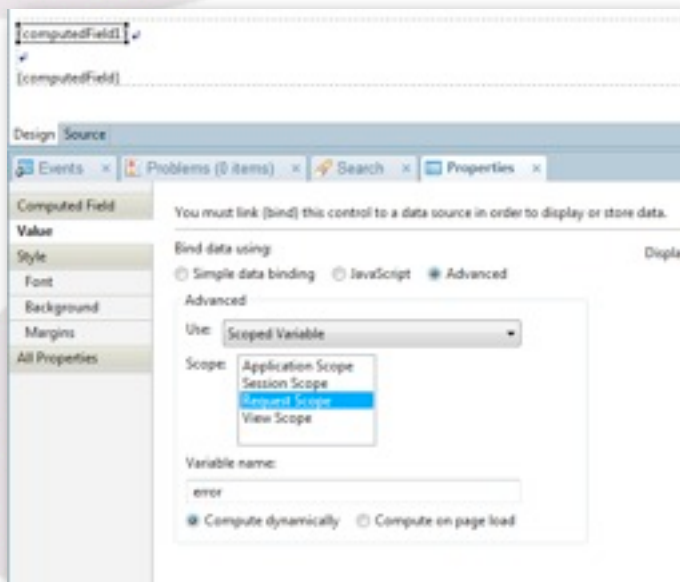
```
1 try{
2   var region = getComponent("region").getSubmittedValue();
3   var list = @DbLookup(@DbName(), "vwRefData", region, 2);
4   if (@IsError(list)){
5     Medusa.getLog().info("There was no region selected");
6     return "Please select a region first!";
7   }else{
8     Medusa.getLog().info("The region selected was: " + region);
9     return list;
10  }
11 }catch (e){
12   Medusa.getLog().error("There was an error: " + e.getMessage());
13   return "Please select a region first (Error)!";
14 }
```



# Error Logging (cont.)

In production, when there's an error you can choose to show your own XPage.

So we have error.xsp that has two computed fields:



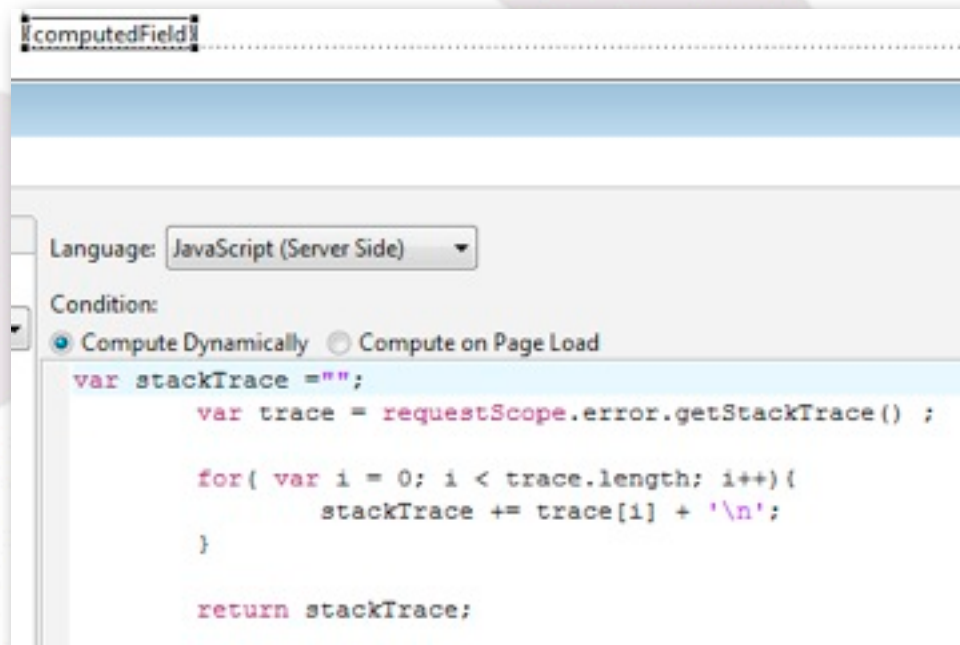
Using a Request Scope variable we can get access to a “hidden” variable called error.

We do this by adding a computed field, and setting the value to Advanced -> Scoped Variable

# Error Logging (cont.)

Then we can make use of that value in another computed field to get the full stack trace (if needed)

To be honest the full stack trace isn't much use.



```
var stackTrace = "";
var trace = requestScope.error.getStackTrace() ;

for( var i = 0; i < trace.length; i++){
    stackTrace += trace[i] + '\n';
}

return stackTrace;
```



**BLUG**  
Belux Lotus User Group



# Demo



# Advanced Validation

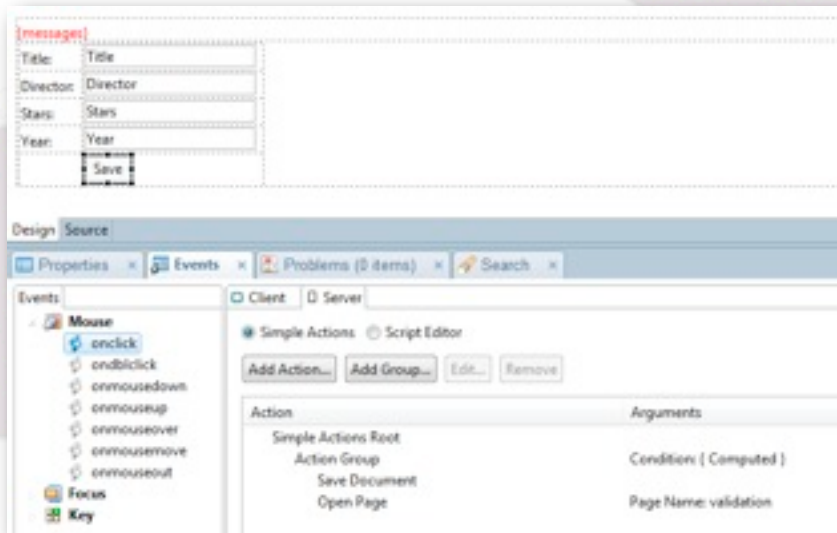
## What we're trying to do

- Perform complex business rules validation
- Take things beyond the basic mandatory checkbox
- Work around the issues that I've had with validating rich text
- Only use this for complex validation, or once you've tried out the "built in" validation options

## How to do it

- Add a computed field with no value for us to put our error messages in (probably make it HTML)
- Build your XPage data binding
- In the Save button properties, make it a normal button (not Submit)
- Add an Action Group to the OnClick which runs your validation
- Return true if the validation is passed
- So that the save action inside the group can take place.

# Advanced Validation (cont.)



In the Action Group of the button's OnClick we call the “validateMovie” function

“validateMovie” implements our business rules and outputs the error messages back into the messages computed field.

Returns false if validation fails so that the actions inside the Action Group are not executed

```
import OpenLogPage;

function validateMovie() {
    try {
        var messages = new Array();
        var doc = document.getElementById("movieForm");
        if (doc.getElementById("title").value == "") {
            messages.push("Please enter the title");
        } else {
            var result = @lookup("@movieForm", "movieForm", doc.getElementById("title").value);
            if (!@isError(result)) {
                messages.push("That title has already been entered");
            }
        }
        if (!@isNumber(doc.getElementById("year").value) || doc.getElementById("year").value.length != 4) {
            messages.push("Please enter a valid year");
        }
        if (messages.length > 0) {
            var out = "Please correct the following errors: <br />";
            for (var i=0; i<messages.length; i++) {
                out += messages[i] + "<br />";
            }
            getComponent("@messages").setValue(out);
            return false;
        } else {
            return true;
        }
    } catch (e) {
        log.LogError(e.getMessage(), SEVERITY_ERROR, e.getMessage(), "", "myValidation", "validateMovie");
        return false;
    }
}
```



**BLUG**  
Belux Lotus User Group

# Demo





# Non HTML Output

## What we're trying to do

- Get some non HTML data from an XPage so we don't have to use agents
  - (e.g. XML, JSON, CSV etc)
- Take advantage of the performance benefits of the JSF servlet engine

## How to do it

- Create a blank XPage and set the entire XPage not to render (in all properties)
- In the “afterRenderResponse” you can get hold of the servlet which is actually driving the XPage and control what it sends back to the browser

```
var exCon = facesContext.getExternalContext();
var writer = facesContext.getResponseWriter();
var response = exCon.getResponse();
response.setContentType("text/xml");
response.setHeader("Cache-Control", "no-cache");
var xml = "<?xml version=\"1.0\" encoding=\"UTF-8\"?><mydata />\n";
writer.write(xml);
writer.endDocument();
```



# Demo



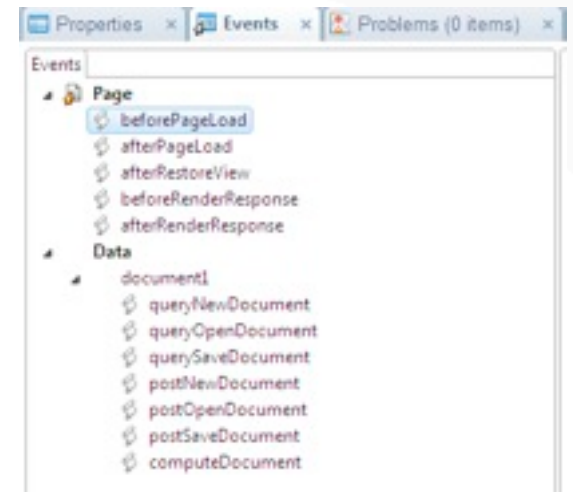
# Data Events (bonus tip)

## What we're trying to do

- Hook onto the same events that we have in the Notes client

## How to do it

- Very very simple, just maybe not so obvious how to get there
- You can use the events to mimic your WebQuerySave event. But I'd highly recommend porting your LotusScript across to Server Side Javascript for performance reasons



# IE8 in 8.5.0 (bonus tip)

- There is a bug in Dojo 1.1.1 (which ships with 8.5.0) which means it won't work with IE8
- To get around this we have to tell IE8 to run in IE7 compliance mode
- To do this add this code to the `beforeRenderResponse` event of your `XPage`:

```
try {  
    if (context.getUserAgent().isIE(8, 8)) {  
        var exCon = facesContext.getExternalContext();  
        var response = exCon.getResponse();  
        response.setHeader("X-UA-Compatible", "IE=EmulateIE7");  
    }  
} catch (e) {  
}
```





# Wrap Up

- XPages can be very simple to use
- But you can do an incredible amount with them as well if you want
- Just pick your level, but it's worth being aware of...
  - partial refreshes are easy if you know the right settings
  - the Dojo framework is huge and powerful, investigate it
  - that Repeats can use data beyond just Views
  - linking SSJS together using import
  - how to merge CSJS and SSJS together
  - facesContext

# Resources

## **Domino Designer Wiki**

- <http://www-10.lotus.com/ldd/ddwiki.nsf>

## **Online Training**

- <http://xpages101.net>
- <http://www.tlcc.com/>

## **Blogs**

- Planet Lotus (of course)
  - <http://planetlotus.org>
- But for XPages especially look at
  - <http://XPagesBlog.com>
  - Stephan Wissel: <http://www.wissel.net/>
  - Declan Sciolla-Lynch: <http://www.qtzar.com>
  - Tim Tripcony: <http://www.timtripcony.com/blog.nsf>
  - John Mackey: <http://www.jmackey.net>
  - Nathan Freeman: <http://www.lotus911.com/nathan/escape.nsf/>
  - Steve Castledine: <http://www.stevecastledine.com>
  - David Leedy: <http://lotusnotebook.com/>

## **The Forums**

- <http://www-10.lotus.com/ldd/nd85forum.nsf>

## **Dojo Site**

- <http://dojotoolkit.org/>

# Any Questions?

Download the samples:

- [mattwhite.me/blug\\_tenpatterns.zip](http://mattwhite.me/blug_tenpatterns.zip)

Contact Me:

- Email: [matt.white@elguji.com](mailto:matt.white@elguji.com)
- Blog: [mattwhite.me](http://mattwhite.me)
- Twitter: [@mattwhite](https://twitter.com/mattwhite)
- Skype: [whitemrx](https://www.skype.com/people/whitemrx)

