

Lotusphere Comes To You 2010



Lotus knows.

Smarter software for a Smarter Planet.



Enhance Your Existing Applications with XPages

Henri Christiaan

*Asia Pacific Techworks - Lotus
IBM Software Group*



Lotus knows.

Smarter software for a Smarter Planet.

This presentation was originally created for Lotusphere
2010 by

Stephan Wissel
and
Steve Castledine



Lotus knows.
Smarter software for a Smarter Planet.

Agenda

- **Know** where you are
 - Client applications vs. web applications
 - Size & Risk of your application
 - Complexity of your code
 - Beauty of your applications
- **Know** what you can do
 - Mapping UI elements
 - UI Considerations
 - Phased approach
- **Know** how to do it
 - Like for like development practices
 - Extra “know” you need
 - Extra “know” to accelerate you forward
- Q & A

Agenda

- Know where you are
- Know what you can do
- Know how to do it

Analyze your application

- How much UI code you you have?
 - Field enter / exit events
 - UI classes
 - Hide When
- Which forms use “Refresh field on keyword change”?
- How much encryption does your application use?
- How complex are your subs and functions?
- How do you use RichText (if any)?

Sample Report*

FORM Quotation Field Count: 1932				
Name	Type	Default	Translation	Validation
custCodeType	editable	"Existing"		
Computed Text	text	custCodeType		
firmCode	editable	@If(@IsError(@DbColumn("": ""; @DbName; "ViewParameter"; 4)); ""; @Subset(@DbColumn("": ""; @DbName; "viewParameter"; 4); 1))		
Computed Text	text	firmCode		
custID	editable	custID		
custName	editable	custName		
custName_1	computed	custName_1		
salesPriceCustomer	editable			
custName_New	editable			
VerifyCode	computed	strMonth := @If(@Month(@Now) > 9; @If(@Month(@Now)=10; "X"; @Month(@Now)=11; "Y"; "Z"); @Text(@Month(@Now))); @If(formStatus="Draft"; @Right(@Text(@Year(@Now)); 1)+strMonth+@Text(@Round(10000-1000*@Random)); VerifyCode)		
Address1	editable			
Address2	editable			

*Reports are available on OpenNTF

Agenda

- Know where you are
- Know what you can do
- Know how to do it

Platform comparison

	IBM Lotus® Notes® Client Application	Classic Lotus Domino® web Application	XPages application
Session	permanent	Time based	Time based
State	Full state information	Stateless	Partially available
Programming	LotusScript JavaScript	LotusScript® JavaScript Java™	JavaScript Java™
UI	RichText	HTML translated from RichText	HTML / Dojo
RichText support	full	weak	partial

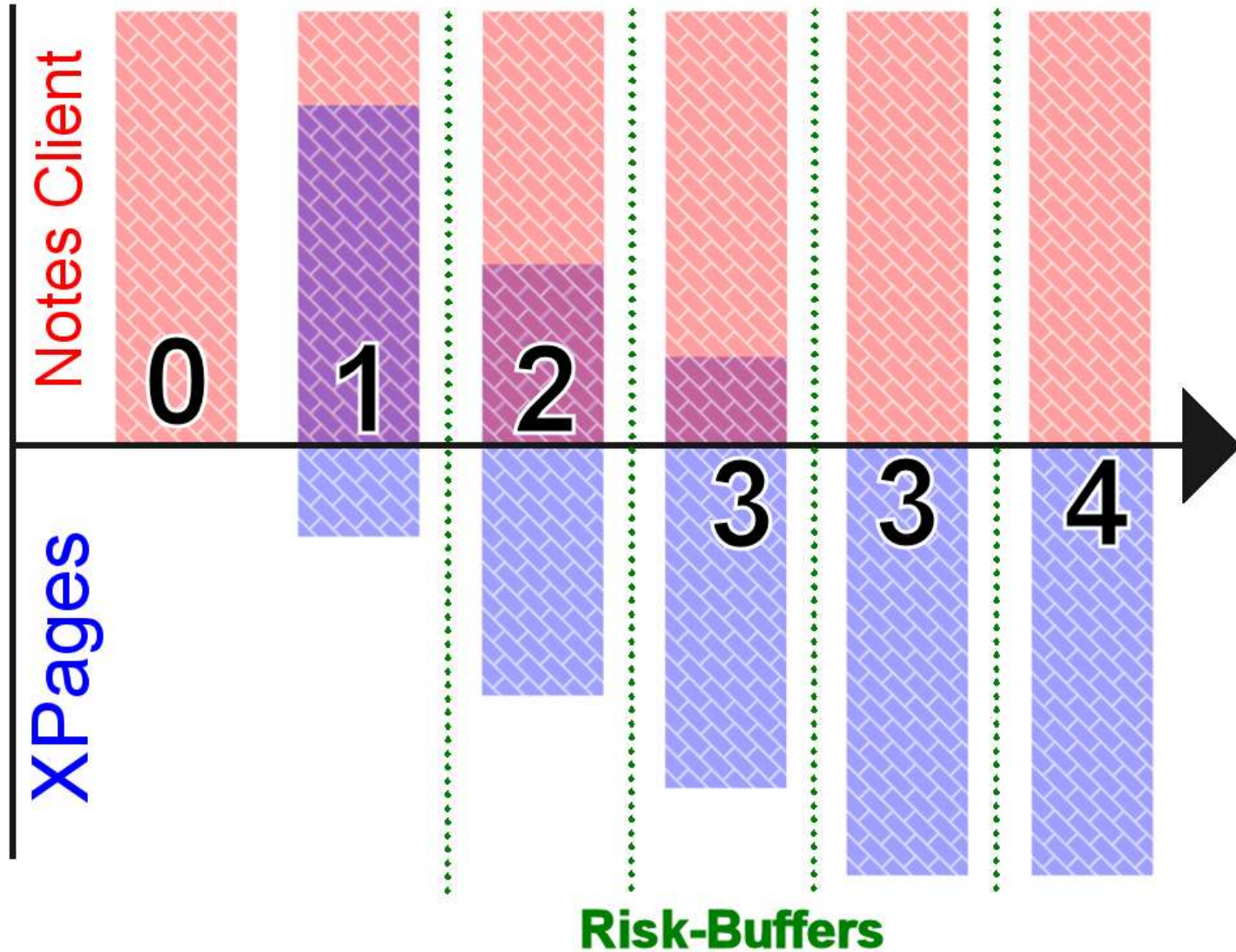
Platform comparison

	Notes Client Application	Classic Domino web Application	XPages application
Update	Full & Partial & Distributed	Full only	Full & Partial (single region)
Encryption / Signature	Yes	No	Under consideration
Layout	Free, typically 2 Frames for views, form Full Screen	Free	Free OneUI recommended
Sections	Standard & Access Controlled	Standard only	Standard (Access controlled under consideration)
Hide when	Per paragraph	Per paragraph + HTML DOM	Per element

Key Questions

- Sequence of enhancements
 - Views first
 - Easy forms
 - Complex forms
 - What UI to pick?
 - 1:1 translation
 - OneUI
 - New corporate look
 - How much refactoring?
 - Good chance to clean-up
 - Refactoring least understood concept for NMU

You need a plan (1)



Agenda

- Know where you are
- Know what you can do
- Know how to do it

OneUI

- XPages has a concept of Themes
- Some out of the box, most notably webstandard and OneUI
- OneUI is a common UI Framework used by IBM Lotus
- Helps to give you a jump start on creating great looking applications
- Examples in Discussion & Wiki Templates
- Also a framework template on OpenNTF.org that does the work for you

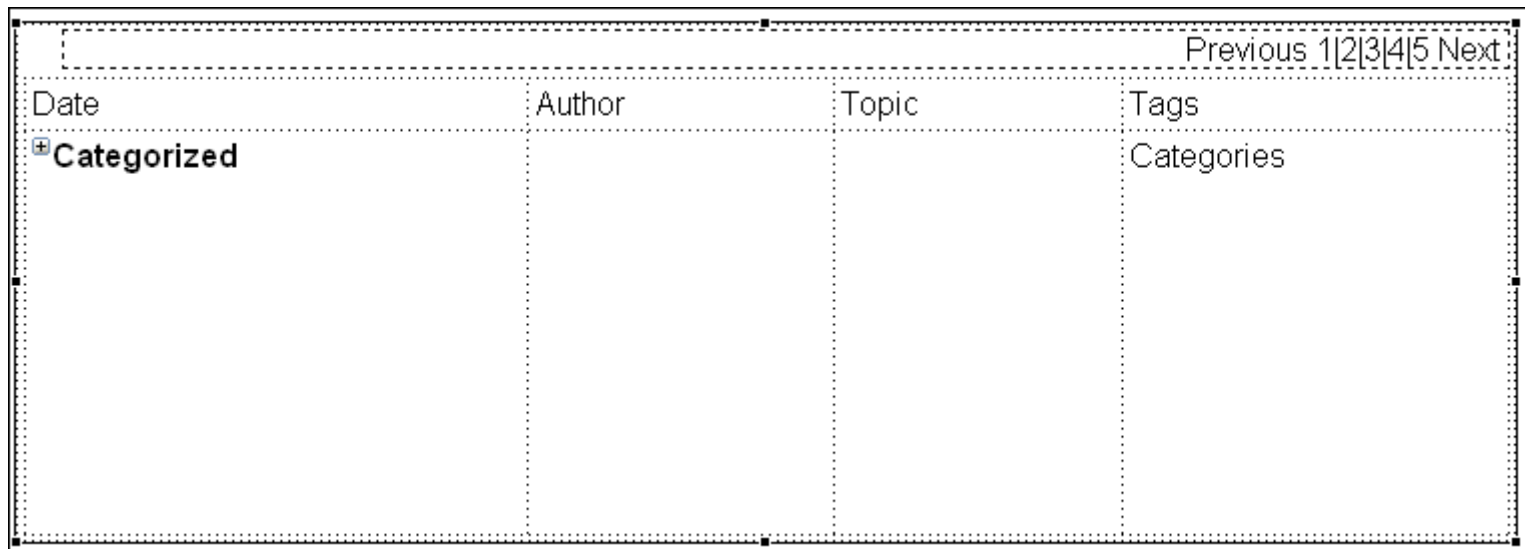


Like for Like Development Practices

- Views (including categorized)
- Action Bars
- Data Forms
- Ability to create, edit or submit these forms
- Sub Forms
- Navigators (maybe using Outlines)
- Sections
- Search
- Business Logic as either LotusScript or Java™

Views

- Three possibilities in XPages
 - View Control (simple and wizard driven)
 - Data Table Control (manually created but tick boxes to add pagers, allows more flexibility)
 - Repeat Control (manually created, maximum flexibility, powerful tool)



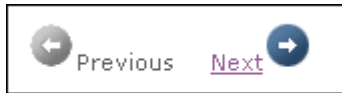
Date	Author	Topic	Tags
Categorized			
			Categories

Views

- For quick easy views the view control is the way
- For things you couldn't do with Lotus Domino® views – repeats are your friend
 - Many rows per document
 - Rich Text possibilities
 - Control everything
- Categorized Views
 - Best to use “View Control”, manual creation in Repeat
 - View control automatically shows category “twistie” for categorised columns as before
 - Is not a property – is read from view design
- Embedded “Filter By Category” Views
 - Can be done via property “categoryFilter” of the view data source
- Upgrade Tip – bookmarks to view links
 - Could exist
 - Currently not an XPage redirect available
 - Live with it or use a Web Site Rule - “Substitution”

View Pagers

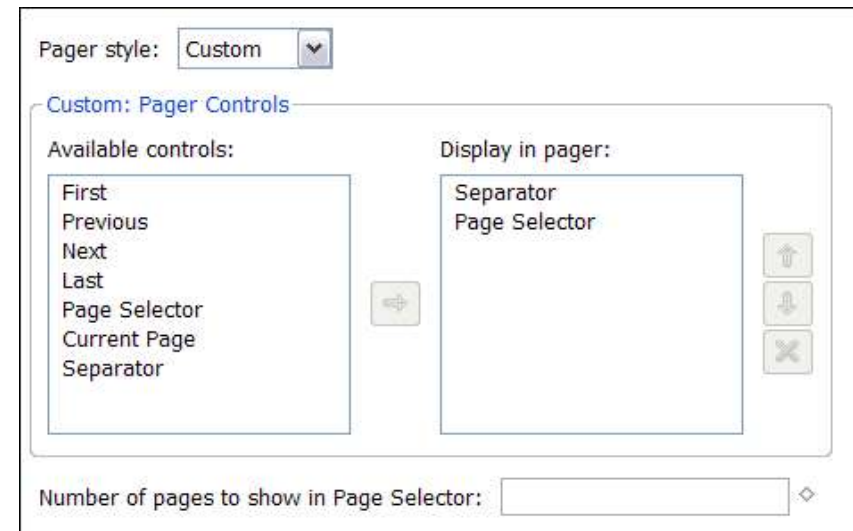
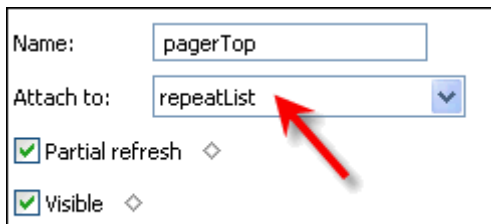
- Domino uses @ Formula based commands for “View Pagers”



- XPages has an easily customisable control to drag and drop, and is added automatically with View Control



- Repeat Controls just need a simple “attach to” field setting in the Repeat Control

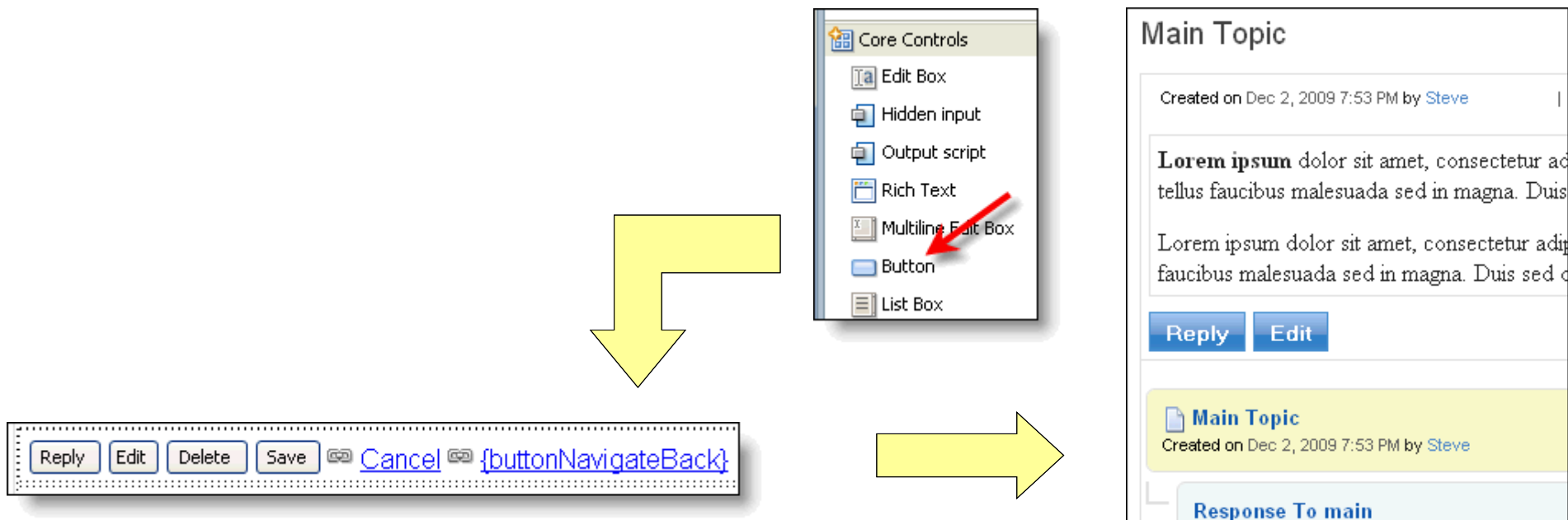


Action Bars

- In Domino you have fixed Action Bars with the ability to add free form buttons in forms

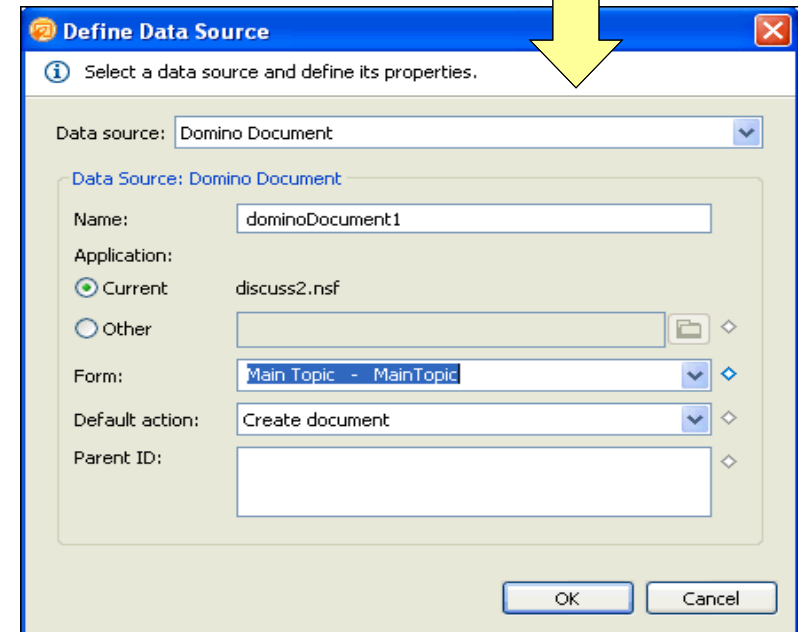
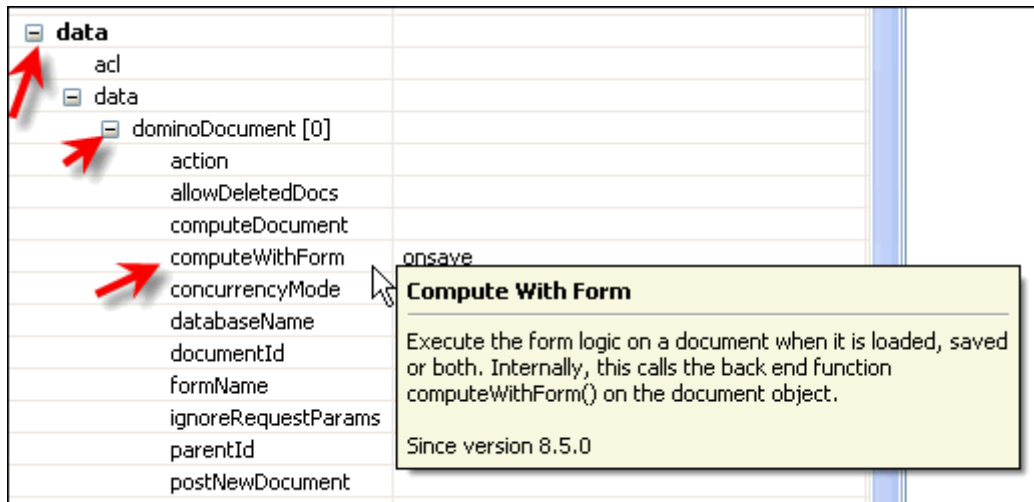
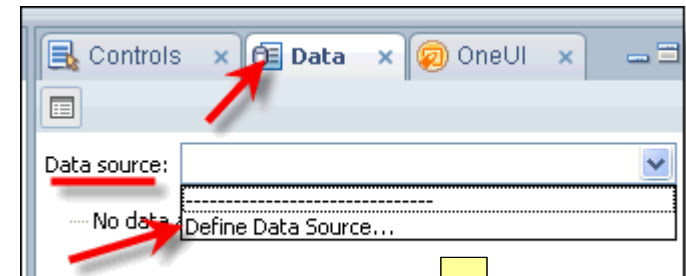


- XPages has “buttons” where can be added where you need them



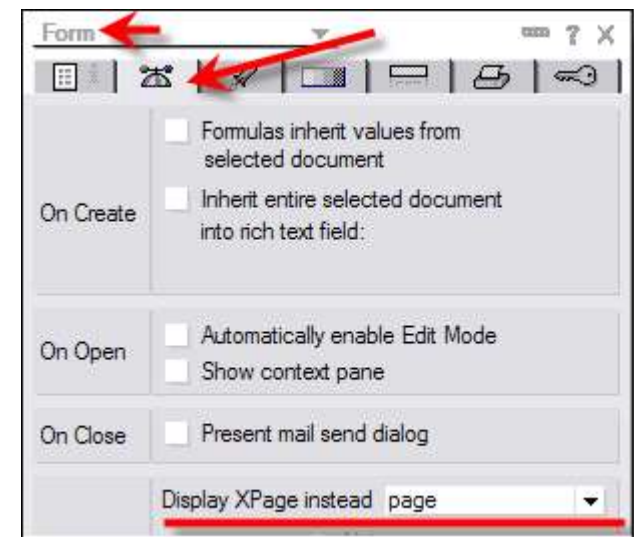
Data Forms

- Existing Notes Forms used for their “schema definition”
- Default values, computations still valid (compute with form etc)
- Connect from XPage via “Data Source”



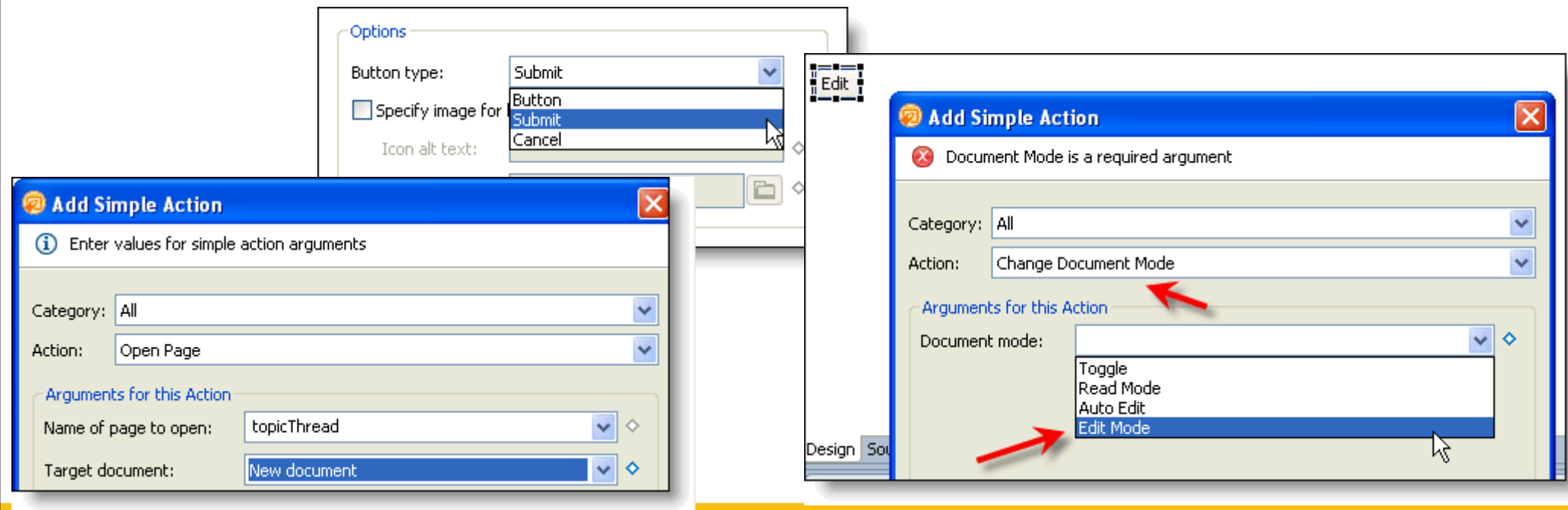
Data Forms

- Upgrade Tip - Getting Existing Documents to launch using an XPage instead
 - If a user has a bookmark to an existing view/document url internally the XPages code will redirect to the XPage url
 - Setting in Form Properties second tab “Display XPage instead”



Create, edit or submit data forms

- Domino uses action bars for buttons and formulas like @Command ([Compose];"form")
- XPages has buttons with simple actions (although code is still possible)
- Data source has a property computeWithForm (onload, onsave, both)

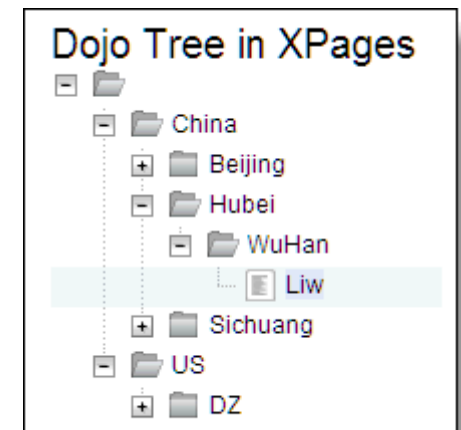


Sub Forms

- Not usable in XPage development
- Recreate those as custom controls for similar reuse
- If they contain fields that are not on forms then those fields are available in XPages, however Designer does not pick them (so manual typing)
- For long term and complex applications consider creating a new form as a clear data schema and include all fields and remove sub forms

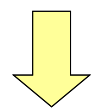
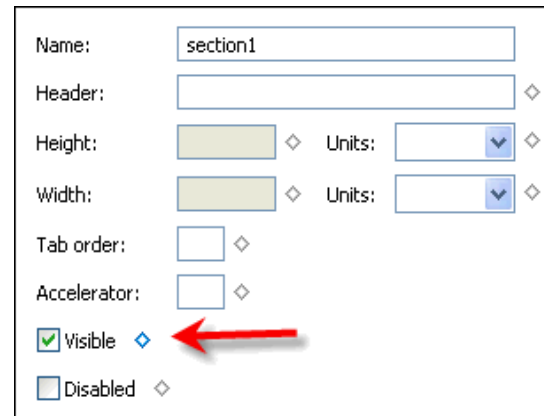
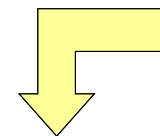
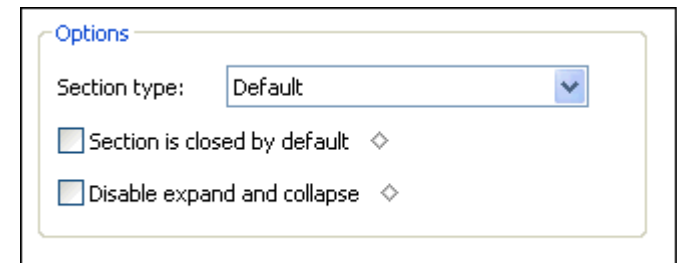
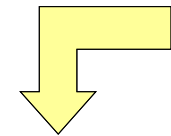
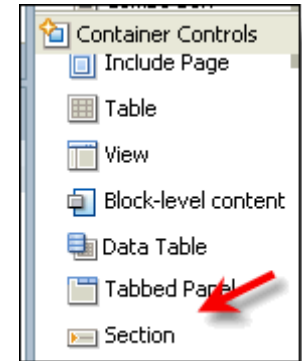
Navigators/Outlines

- Outlines currently have no “out of the box” automatic replacement
- However this is web development so many easy examples available to reuse
- OpenNTF.org has a few controls for download. The discussion and wiki template's use these



Sections

- Notes development has sections
 - Standard
 - Controlled Access
- XPages also has sections via the section Container Control
- Similar properties to Notes
- Can be “Controlled Access” via the “visible” property



Search

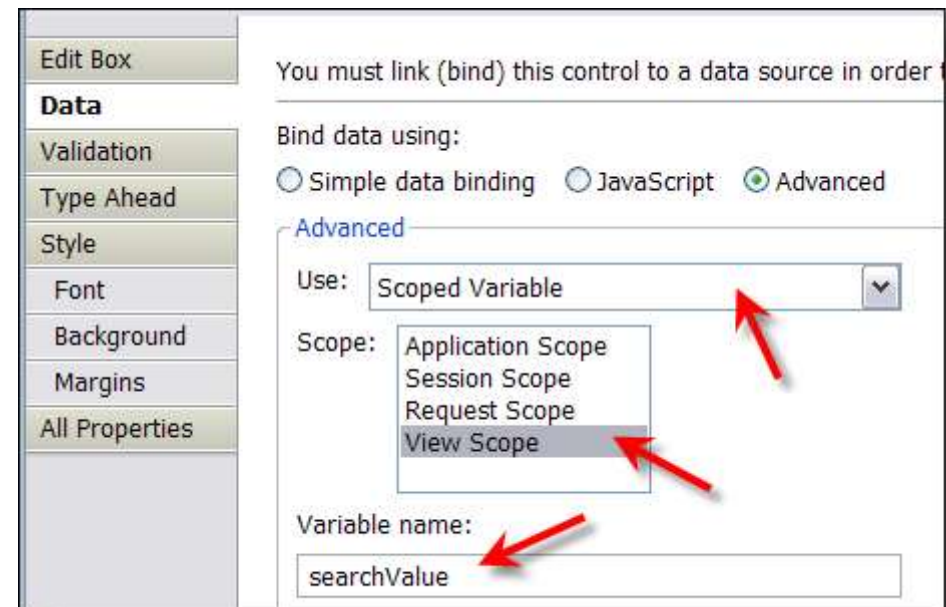
- Domino development has built in search and results form (together with custom built options via \$\$SearchTemplate etc)
- XPages offers simple inline search or custom built search query and results custom controls/XPages (but built yourself or downloaded from openntf.org)
- The concept is just adding a Search query to the “data source” in use for that XPage
- Give the search property a value, say collect something from the url - “param.searchValue”
- Optional parameter using “searchMaxDocs” to restrict number of results returned for search

[-] data	
acl	
[-] data	
[-] domoView [0]	
categoryFilter	
databaseName	
dataCache	full
expandLevel	# if(sessionScope.ec==
ignoreRequestParams	
keys	
keysExactMatch	
parentId	
postOpenView	
queryOpenView	
requestParamPrefix	
scope	
search	# param.searchValue
searchMaxDocs	



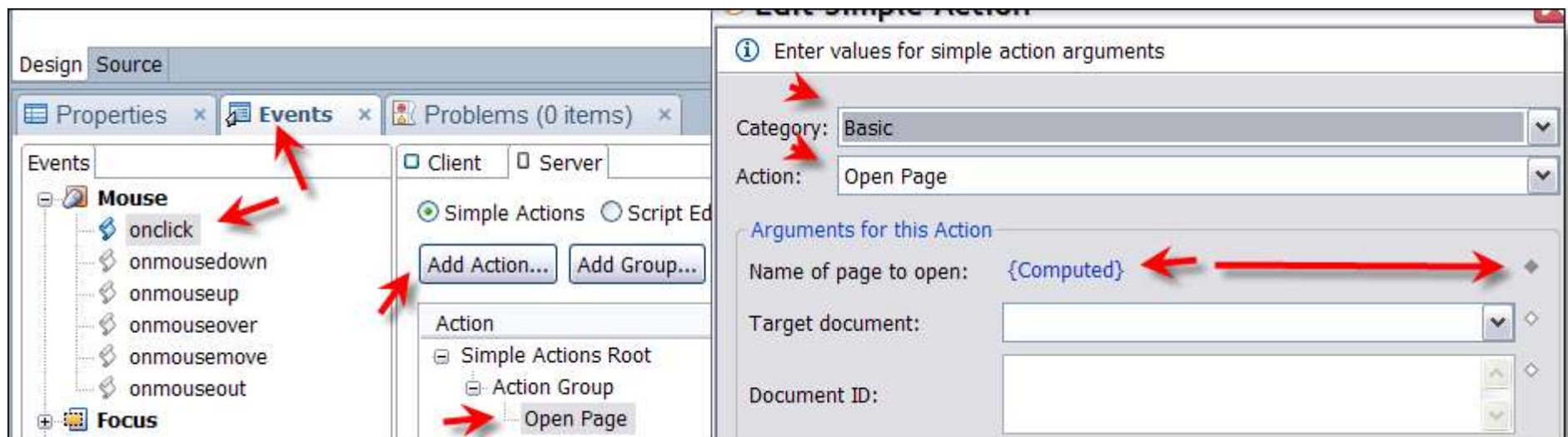
Search

- Then supply an Edit Box to take search parameter
- In Data properties link to “Scoped Variable” - “View Scope” and give data a name (this then places the data in memory to be picked up by search action Link)



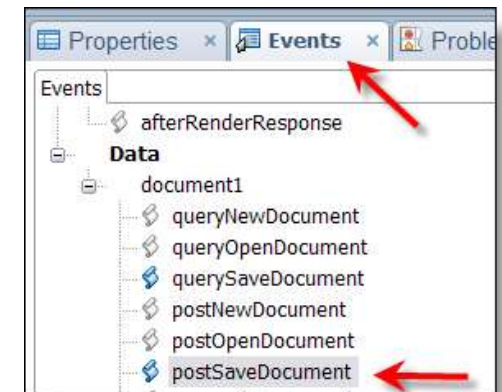
Search

- Lastly add a Link control to submit search request
- Using either a computed Link or onClick Event compute the next page or url and retrieve search value using “viewScope.searchValue”.
- Example computed url:
 - “xpage.xsp?searchValue=” + viewScope.searchValue



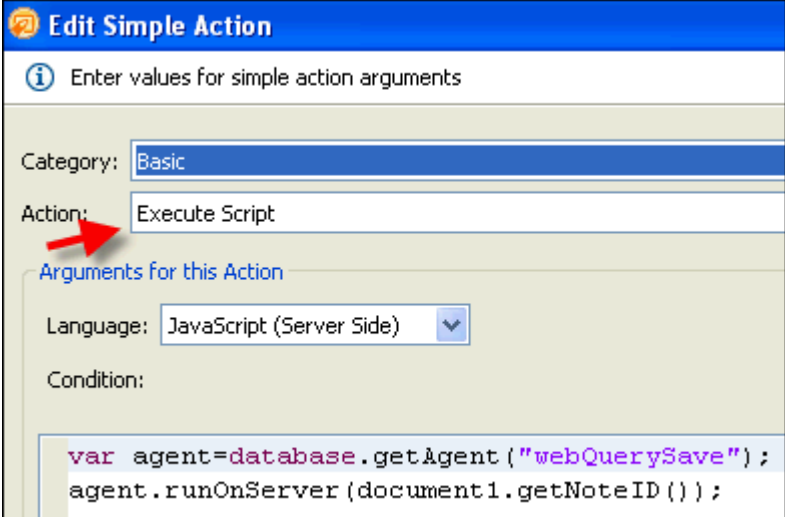
Business Logic

- Existing Logic maybe in Lotuscript or Java Agents/Libraries
- Default Language for XPages is Server Side JavaScript (SSJS)
- SSJS supports the full Notes Domino object model like LotusScript
- You can use @Formula in that language but not LotusScript
- What to do? agent.RunOnServer(docid) on postSaveDocument event
- This will run your agent, send it the id of the to be processed document
- XPage code will resume once the agent has finished
- Agent needs to retrieve its own “context” document using the id passed
- `doc = db.GetDocumentByID(agent.ParameterDocID)`



Business Logic

- Your choices:
 - Keep as is – will work fine via previously described method (improvements planned for maintenance releases)
 - Rewrite using SSJS
 - Bigger/More advanced stuff – maybe rewrite in java and this can be used direct from SSJS or perhaps at more advanced level like managed java beans
- Note from the code examples that the SSJS looks similar to LotusScript



Edit Simple Action

Enter values for simple action arguments

Category: Basic

Action: Execute Script

Arguments for this Action

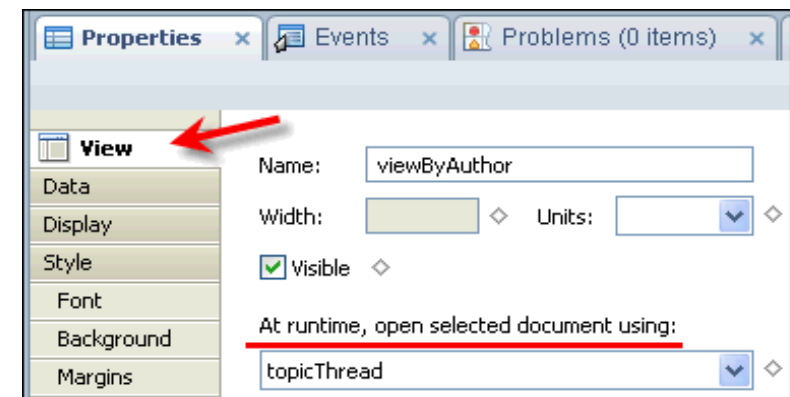
Language: JavaScript (Server Side)

Condition:

```
var agent=database.getAgent( "webQuerySave" );
agent.runOnServer( document1.getNoteID() );
```

Multiple NSF's

- Some existing applications may span multiple nsf's
- Maybe for different entry points but mainly as data silo's
- XPages makes it easier to retrieve and combine this data on one XPage (do not read this as joining views across nsf's – presentation only)
- Good practice maybe would be to combine XPage design in one nsf
- XPages in the Client currently has a security restriction launching an XPage in “another” nsf
- Key tip here is when you create views and you want to launch a specific XPage, use the column property “open selected document using” to select that XPage



Extra “know” to accelerate you forward

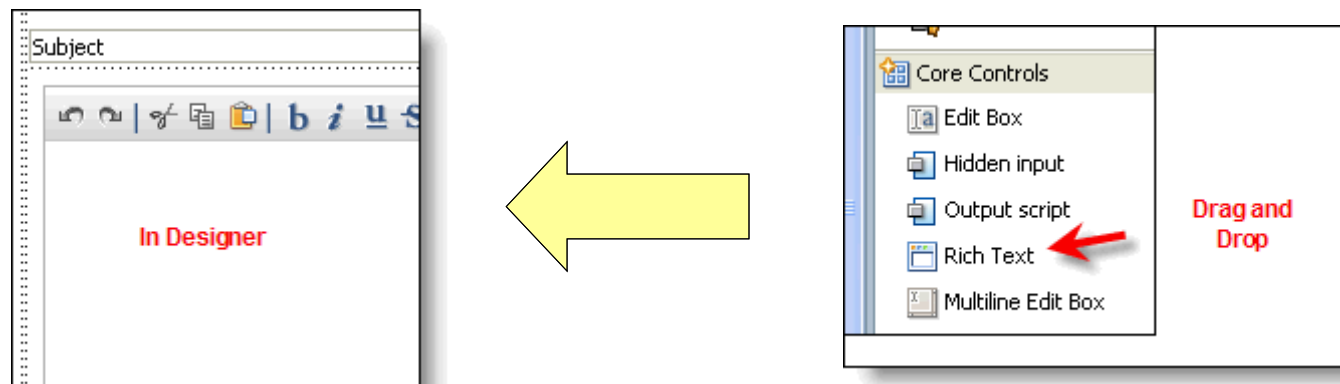
- Rich Text
- Type Ahead
- Field Validation
- Date Time Pickers
- Linked Combo Boxes
- File Upload
- Dojo Widgets
- Tabbed Tables
- OneUI
- XSS Protection
- Relational Database Access
- OpenNTF

Rich Text

- XPages improves rich text handling, a lot
- Out of the box Dojo Rich Text Control



- Automatic Filters for XSS
- Easy use of a “Custom Convertor” to parse results if you choose (wiki markup perhaps)



Type Ahead

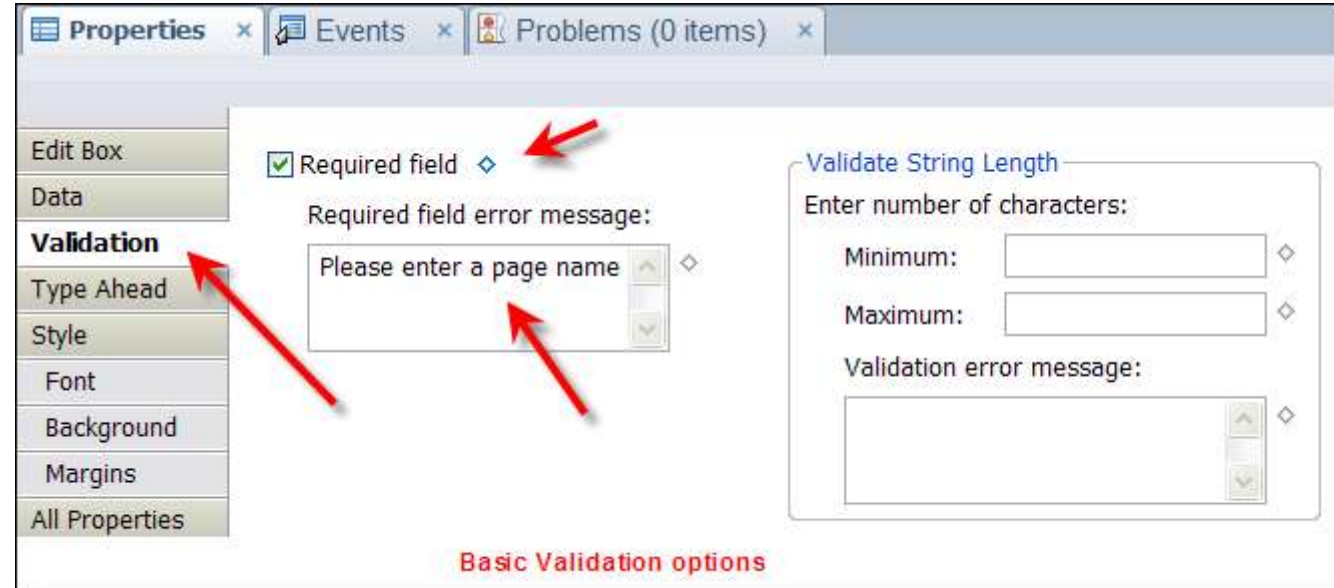
- Simple to turn on for any field – just “Enable Type Ahead” in Type Ahead property box
- Then supply a Notes view to get the suggestions from
- Fine tune Mode (partial, full), separator, minimum characters

The screenshot shows the 'Page Information' section of a Lotus Notes document. The 'Category' field is empty, and the 'Tags' field contains the letter 'D'. A dropdown menu is open below the 'Tags' field, showing a list of suggestions: 'database', 'DCT', 'Deployments issues', 'Domino', 'domino blog', 'dominoblog.ntf', and 'dx tags'. A red arrow points to the 'Type-ahead' label above the dropdown. Below the suggestions list, there are radio buttons for 'Compute Dynamically' (selected) and 'Compute on Page Load'. A text box contains the formula: `@DbColumn("", "vTags", 2);`. A red arrow points to this formula with the text 'Standard Notes formula for Lookup'. A yellow arrow points from the 'Type Ahead' property box in the bottom-left screenshot to this formula box.

The screenshot shows the 'Type Ahead' property box in the Lotus Notes interface. The 'Enable Type Ahead' checkbox is checked. The 'Mode' is set to 'Partial'. The 'Suggestions' field contains the text '{Computed}'. A red arrow points to the 'Type Ahead' tab, and another red arrow points to the 'Suggestions' field.

Field Validation

- Enabled within properties for each field
- From simple “required” field to conditional and string length checking
- To enable basic validation go to “Validation” tab of Edit Box properties



Field Validation Advanced

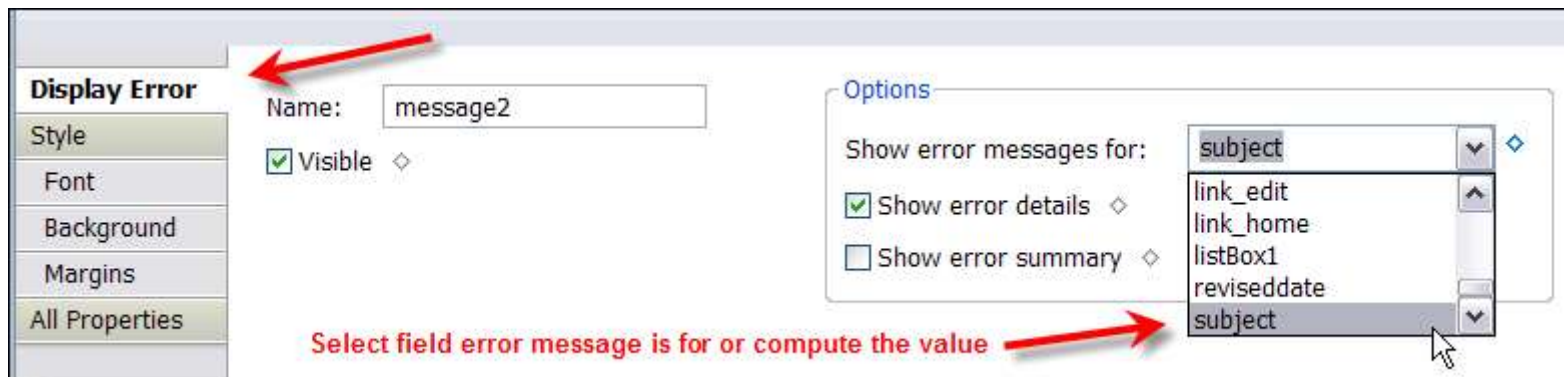
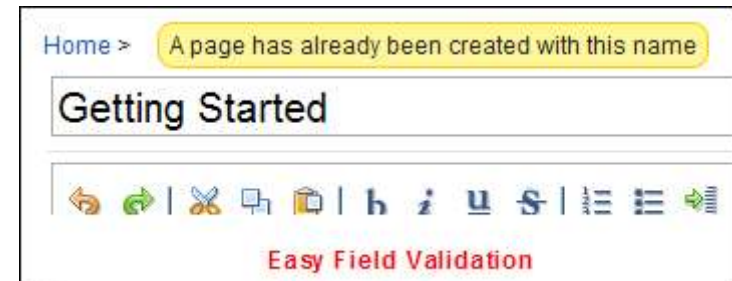
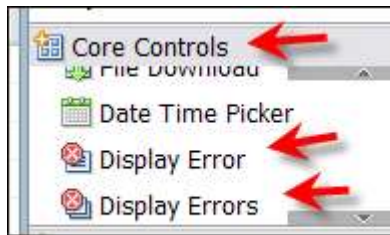
- More advanced use – add a “validator” via “All Properties”
- Can add as many validators as you like
- Flexible options – adding a “customValidator” allows you to do things like checking for duplicate field value

The screenshot shows the 'All Properties' tab in the Lotus Notes properties window. The 'validators' section is expanded, showing a list of validators. A red arrow points to the 'validators' header, and another points to a custom validator entry labeled 'Duplicate name validator (customValidator)'. The 'expression' field for this validator contains the text '//check for page ...ion if new doc...'. The 'message' field contains the text 'A page has already been created wit...'. A red arrow points to the 'message' field with the text 'Message passed to user'.

```
//check for page name duplication if new doc
if (document1.isNewNote()) {
var view:NotesView=database.getView("vContent3");
var doc:NotesDocument=view.getDocumentByKey(value,true);
if (doc==null) {
return true;
}else{
return false;
}
}else{return true}
```

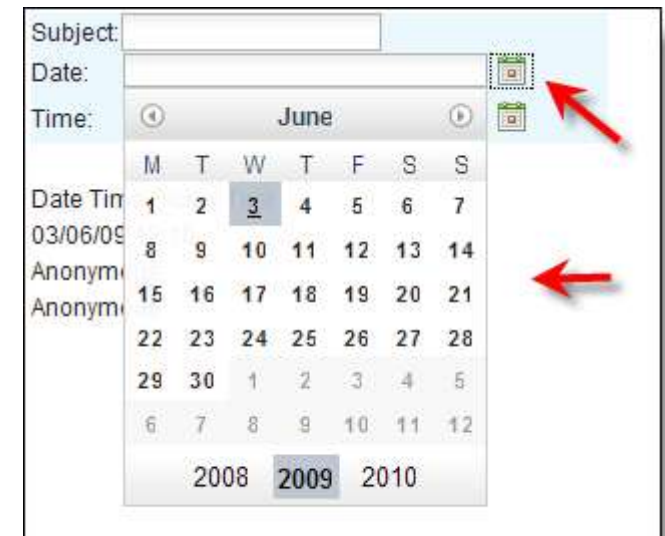
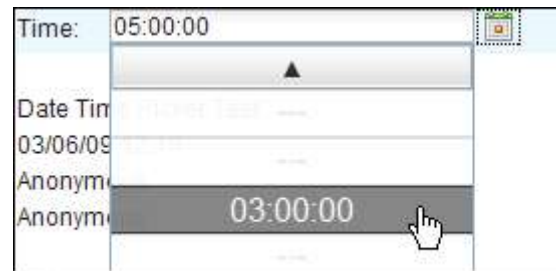
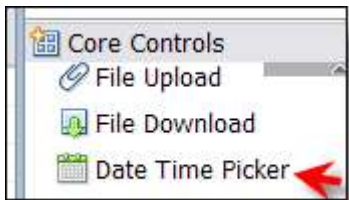
Field Validation – displaying message to user

- You use Core Control “Display Error” or “Display Errors” to position where the user sees the error when they submit the data.
- Simple drag the control to where you want it to display
- For “Display Error” - just link it to the field that has the validation



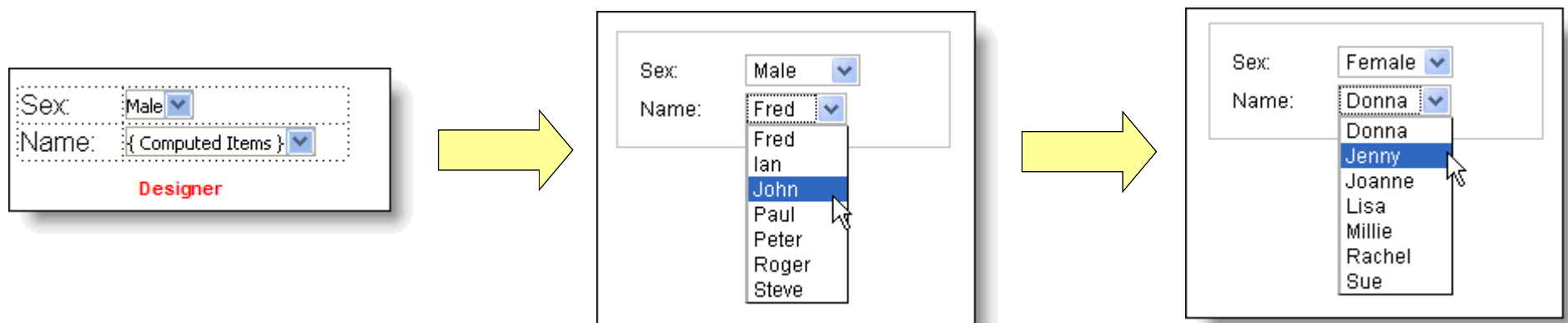
Date time Picker

- Core control – just drag and drop onto your XPage – no coding required
- Just a standard edit box linked to a date/time field with property “Use date/time picker pop-up” set



Linked Combo Boxes

- Options in a second combo box are dependent on the value selected in the first one
- Create two Combo Boxes (first with manual values if required)
- First Combo set Partial Update and id of second Combo, also set “do not validate”
- Second Combo add a “computed item” for values lookup first combo and use for filter for a @DbLookup



Linked Combo Boxes

Client Server

Simple Actions Script Editor

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

Action	Arguments
Simple Actions Root	

Server Options

Full Update

Partial Update Select Element...

IDs: Name

No Update

No Submission

Do not validate or update data

Set partial execution mode

First Combo Box Settings

Design Source

Properties Events

Combo Box

Data

Values

Style

Font

Add Item Add For

Label

{ Computed Items }

Language: JavaScript (Server Side)

Condition:

Compute Dynamically Compute on Page Load

```

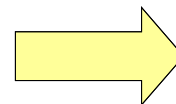
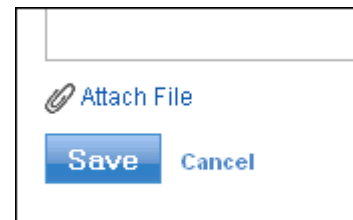
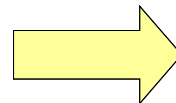
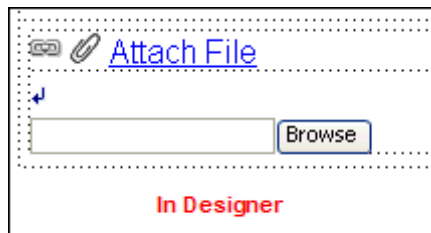
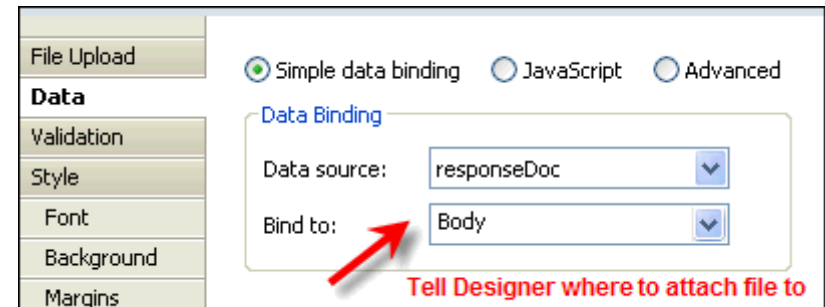
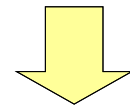
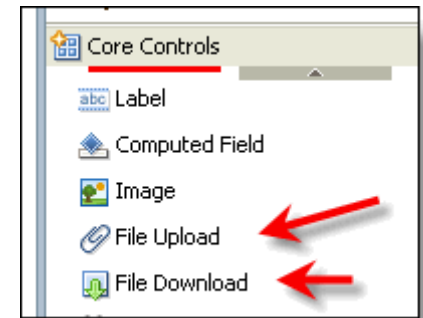
1 var sex=getComponent ("sex" ).getSubmittedValue ();
2 var names=@DbLookup (@DbName (), "vNamesBySex", sex, 2 );
3
4 return names;

```

Second Combo Box Computed Values

File Upload (and download)

- Drag and Drop Core controls
- Just need to specify which field to attach the files to
- Customizable



File Type	Size	File Name	Created On
image/gif	1 KB	1.gif	12/18/09 4:45 PM

Dojo Widgets

- As well as the out of the box dojo core controls you can use others
- Such as Dialog, Tree, Progress Bar, etc (www.dojocampus.org)
- Need to add `dojoParseOnLoad=true` & `dojoTheme=true` to XPage
- Then add the dojo module resource you need eg “dijit.Dialog”
- XPage elements have a `dojoType` & `dojoAttributes` which you use

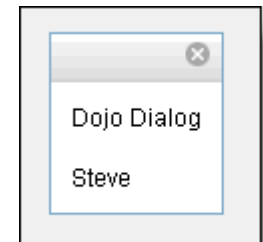
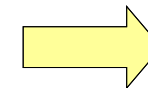
Property	Value
basics	
binding	
createForm	
dir	
dojoParseOnLoad	true
dojoTheme	true
enableModifiedFlag	

rendererType	
resources	<input type="button" value="+"/> <input type="button" value="-"/>
dojoModule [0]	
basics	
condition	
name	dijit.Dialog
rendered	<u> </u>

Add resource "dojoModule"

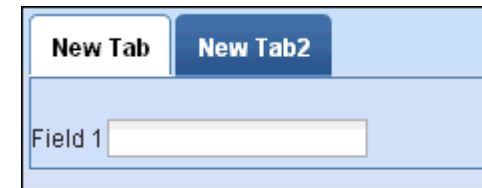
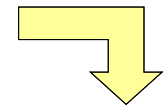
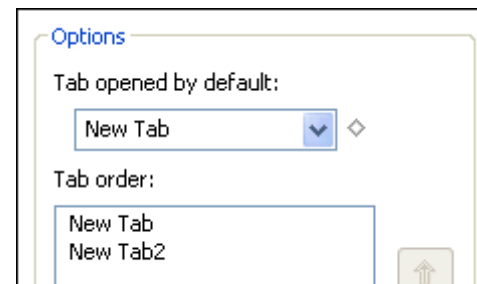
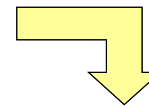
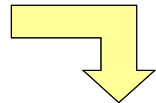
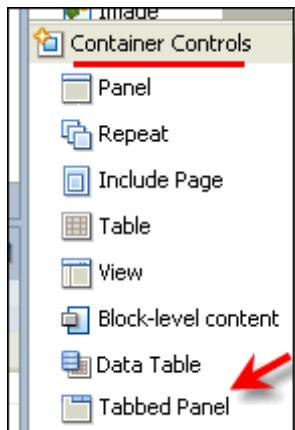
dojo	
dojoAttributes	
dojoType	dijit.Dialog
styling	

Panel Properties



Tabbed Tables

- XPages has a drag and drop core control “Tabbed Panel”
- Very simple configuration
- Drag and Drop the other controls etc you wish to be on each tab



Relational Database Access

- Not supported natively by XPages at this point
- However not difficult to surface data from different back end data sources via JDBC
- You can access via JDBC direct from your SSJS
- Better option is to build java classes to handle the access and call these from SSJS - importPackage(package)
- Maybe even use the Java Bean capability that comes with JSF part of XPages

```
var driverClass:java.lang.Class;
var driver:java.sql.Driver;
var connection:java.sql.Connection;
var statement:java.sql.Statement;

driverClass = java.lang.Thread.currentThread().getContextClassLoader().loadClass(
"org.gjt.mm.mysql.Driver");

driver = driverClass.newInstance();

java.sql.DriverManager.registerDriver(driver);
connection = java.sql.DriverManager.getConnection("jdbc:mysql://dataserver.mycompany.com/
tours?user=datauser&password=datapw");

statement = connection.createStatement();
```



Q & A

Resources

Open Source Code for Notes & Domino	www.OpenNTF.org
Domino Designer (& XPages) wiki	www-10.lotus.com/ldd/ddwiki.nsf
One UI Documentation	www-10.lotus.com/ldd/ddwiki.nsf/dx/12102009164524SCALK8.htm
One UI Framework Project	www.openntf.org/Projects/pmt.nsf/ProjectLookup/XPages%20Framework
Steve Castledine Blog & Twitter	www.stevecastledine.com Twitter:scastledine
Stephan Wissel Blog	www.wissel.net
Lotus Blogs	www.planetlotus.org
Using XPages to Integrate Between Notes Applications with Relational Databases	www-10.lotus.com/ldd/ddwiki.nsf/dx/xpage_integration_rdb.htm

Thank You



Lotus knows.

Smarter software for a Smarter Planet.



Legal Disclaimer

© IBM Corporation 2009. All Rights Reserved.

The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

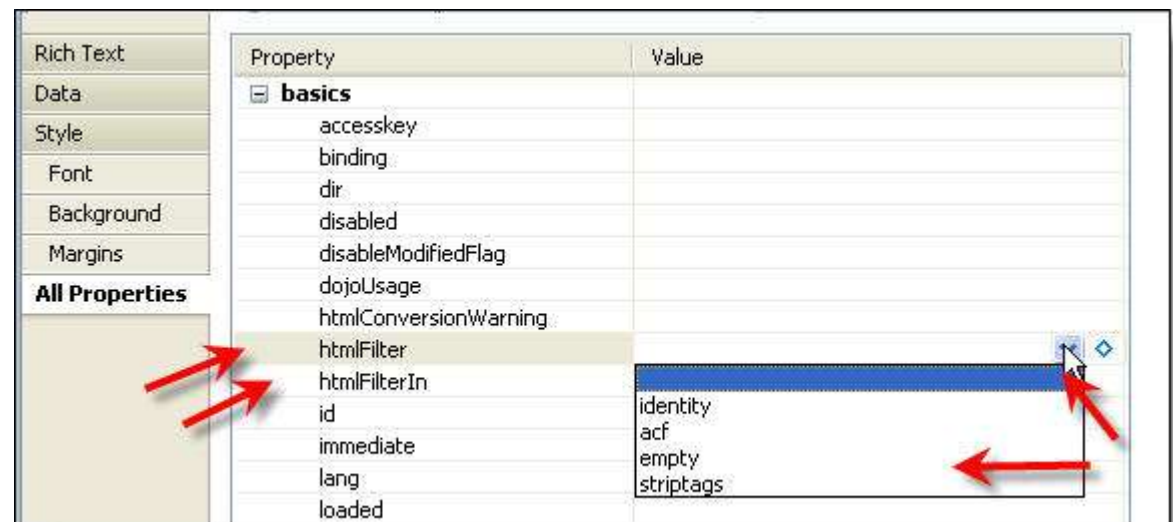
References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Lotus Notes, Domino,

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

XSS Protection

- Active Content Filtering (ACF)
- Pure Java open source engine bundled with XPages/Domino
- Automatically Filters nasty html or javascript embedded within rich text or fields
- Switch on via All Properties and “htmlFilter” or “htmlFilterIn”
- Setting “acf” will do a full filter, “striptags” will strip html tags only



Translate Classic Notes to XPages

Notes artifact	XPages artifact
Notes Form	XPage and a serverside JS library with a governing JS object, Custom Control
Notes Subform	Custom control
Notes View	XPage with embedded view

Translate Classic Notes to XPages

Notes artifact	XPages artifact
View Actions	Custom control with unordered list (ul) of action links, server side JS library
Notes fields	Controls with data binding
Notes tables	Table control
Notes tabbed table	Tab control in conjunctions with content panels
Hide when formula	visibility formula in conjunction with a page wide governing object

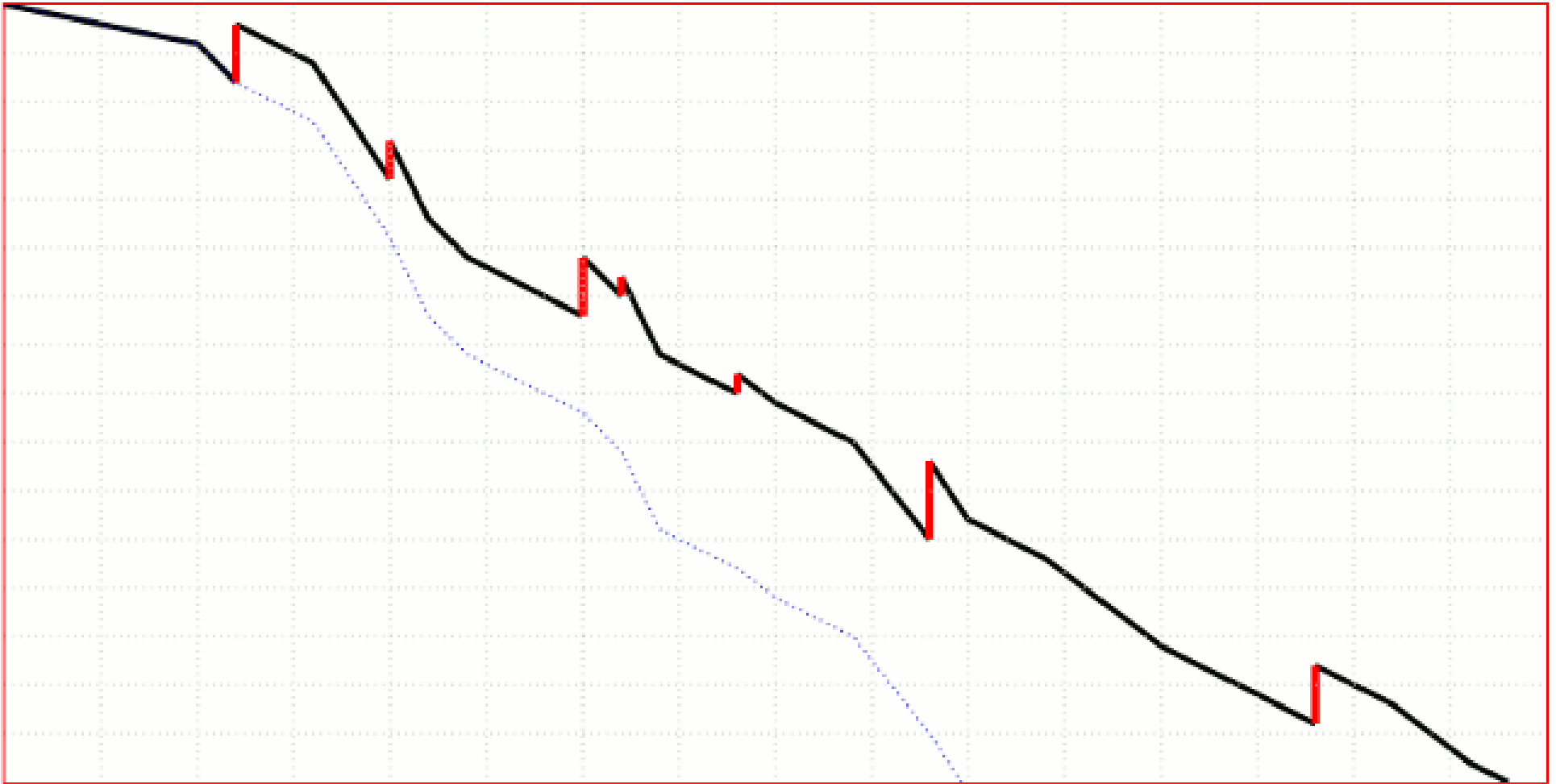
Translate Classic Notes to XPages

Notes artifact	XPages artifact
Single panel table	accordion container
Section	combination of 2 panels and 2 images and an action to expand/collapse (Dojo)
Text fields	Input boxes or Dijit Text Area.
Access controlled section	custom control build of 3 panels
Button, Color pick.	Button, Dijit Color picker
Lotus Script client	JavaScript functions

Translate Classic Notes to XPages

Notes artifact	XPages artifact
RichText	Mime Based RichText
Names (+Au, +Re)	Custom control with name picker
Framesets	Dropped for Ajax UI
Outlines	Lists of links styled with CSS
Embedded views	views in a XPage
Pages	XPages
Agent	Agents called from a XPage event

You need a plan (2)



Advanced Search

- Show/Hide parts of your view depending on whether a search parameter is detected – for things like “Searching For: Term”
- Basic and Advanced Search custom Controls available on OpenNTF
- They plug into your application and obviously code is open to customize

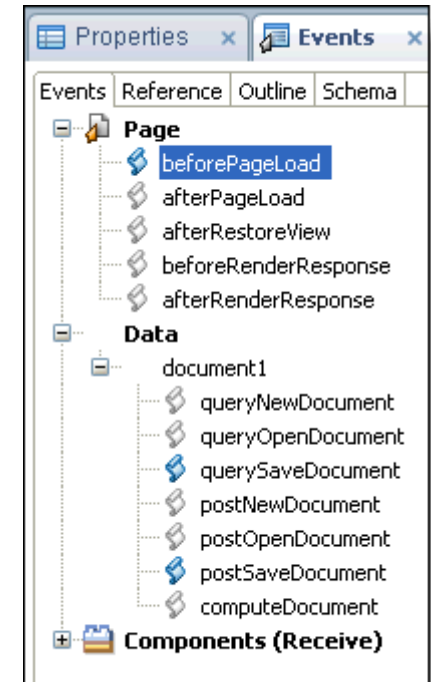
Find documents with

all of the words	<input type="text" value="Niklas"/>	All your words, in any order.
the exact phrase	<input type="text"/>	The exact words in the order you entered the
at least one of the words	<input type="text"/>	One or more of your words. This broadens y
none of the words	<input type="text"/>	Any of these unwanted words
boolean search	<input type="text"/>	Refine your search query using operators, fi
Author	<input type="text"/>	The author's first and/or last name.
Date:	<input type="text" value="anytime"/> ▼	Narrow your search to a specific timeframe
Tag:	<input type="text" value="any tag"/> ▲ ▼	Narrow your search to a specific tag or tags

 Search

Web Query Open / Web Query Save

- Your favourite question – it seems
- WQO/WQS agents are only usable via the previous method described
- XPages themselves have a full array of events that take over from these making them more flexible
- Range from data events that match those available in existing Notes development
- To XPage “lifecycle” events like beforePageLoad, afterPageLoad etc

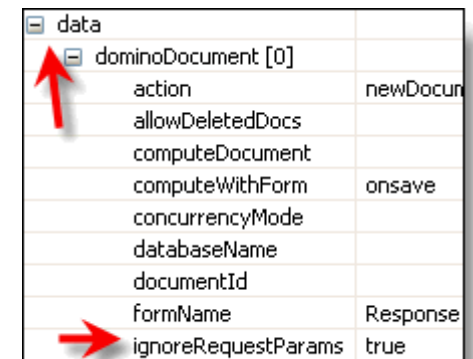


Extra “know” you need

- Rich Text
- Multiple Data Forms (ie responding to other documents)
- \$\$ViewTemplate's
- \$\$ErrorTemplates
- Multiple NSF's

Multiple Data Forms

- XPages allow multiple data sources on one page
- 1 Form 1 View, 1 Form 2 Views, 2 Forms 1 View etc etc
- You can link them together to display depending on other selections
- Views are simple, Forms need some thought
- The “master” Form is controlled via the XPages model
- Subsequent Forms you have to drive and “unlock” from the master
- Use data source setting “ignoreRequestParams=true” to do this



data		
dominoDocument [0]		
action		newDocun
allowDeletedDocs		
computeDocument		
computeWithForm		onsave
concurrencyMode		
databaseName		
documentId		
formName		Response
ignoreRequestParams		true

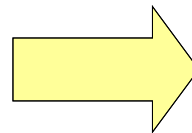
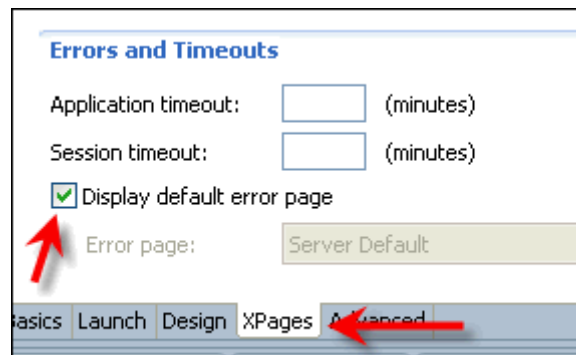
\$\$ViewTemplate's

- Domino has \$\$ViewTemplates and we can provide one design element to “style” how all views look within a browser
- XPages doesn't have this concept however there are other design patterns available
 - 1 Custom control with custom properties determining view
 - Columns must be identical though
 - Use a custom control as an overall container for your site and add an “Editable Area” to the view control



\$\$ErrorTemplates

- Errors used to be caught by \$\$ErrorTemplates
- XPages by default gives you Error 500
- Turn on error page via App Properties > Xpages > Display default error page
- But gives an error to the user
- So we can also select a pre created Error Page



Error source

Page Name: /e.xsp
Control Id: computedField1
Property: value

Exception

Error while executing JavaScript computed expression
Reference error: 'varName' not found

Javascript code

```
1: varName
```

► **Stack Trace**

\$\$ErrorTemplates

- With a custom Error XPage you can send a friendly message to the user and send the real error to a log
- You add a computed field and either enter requestScope.error in the JavaScript editor or add a request scope variable called error
- The 'error' variable contains the actual Java Exception object
- Using java.lang.Throwable methods, you can create a more detailed error message

Errors and Timeouts

Application timeout: (minutes)

Session timeout: (minutes)

Display default error page

Error page:

Bind data using:

Simple data binding JavaScript Advanced

Advanced

Use:

Scope:

Variable name:

Compute dynamically Compute on page load