The Iam Lotus User Group

Using XML and RDBMS Data Sources in XPages

Paul T. Calhoun
NetNotes Solutions Unlimited, Inc

The Iam Lotus User Group

IBM

panagenda

GSX Groupware Solutions

teamstudio
Get the most from Notes®

MartinScott

IamLUG 2010 Sponsors

IamLUG

Speaker Information

- Independent
  - Consultant, Trainer, Mentor, Speaker for past 18 years
- Location
  - Houston, Texas
- Age
  - Older than some, not as old as he looks
- Married / 4 Kids (Which explains the grey hair) / 1 Dog / 1 Cat
- Hobbies
  - Riding Motorcycles, Speaking at LUG's, Cooking
- Certifications
  - IBM / Lotus / Rational Developer / DB2
    - Administration and Developer certified in every release since release three

What We'll Cover …

- Producing XML from Domino
- Transforming XML as XPage Content
- Parsing XML as XPage Content
- RDBMS Data in an XPage using ZERO JAVA CODE !!!
- RDBMS Data in an XPage using JDBC
- Wrap-up/Summary/Questions

What We'll Cover …

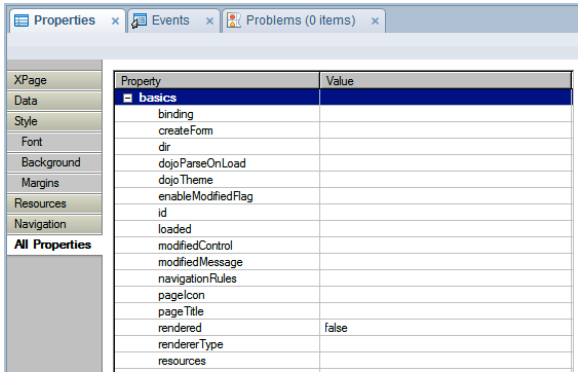
- Producing XML from Domino
- Transforming XML as XPage Content
- Parsing XML as XPage Content
- RDBMS Data in an XPage using ZERO JAVA CODE !!!
- RDBMS Data in an XPage using JDBC
- Wrap-up/Summary/Questions

Producing XML From Domino Data

- Domino can produce XML in any of the following ways
  - ?ReadViewEntries
  - Form Design
  - View Design
  - Notes Document
  - Agents
  - Servlets
  - XAgents (XPages implemented as Agents)
    - This is the option we will use
- Bottom line is it doesn't matter where the XML comes from as long as it's
  - Accessible via URL
  - Well-Formed

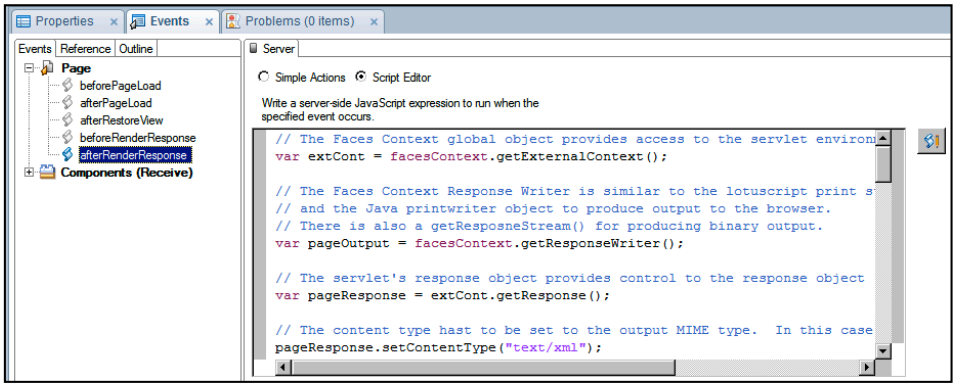
XML via an XAgent

- **An XAgent is an XPage...**
  - **That has it's rendering property set to false**



XML via an XAgent

- **An XAgent is an XPage...**
  - **That has all of the content produced in the afterRenderResponse page event**

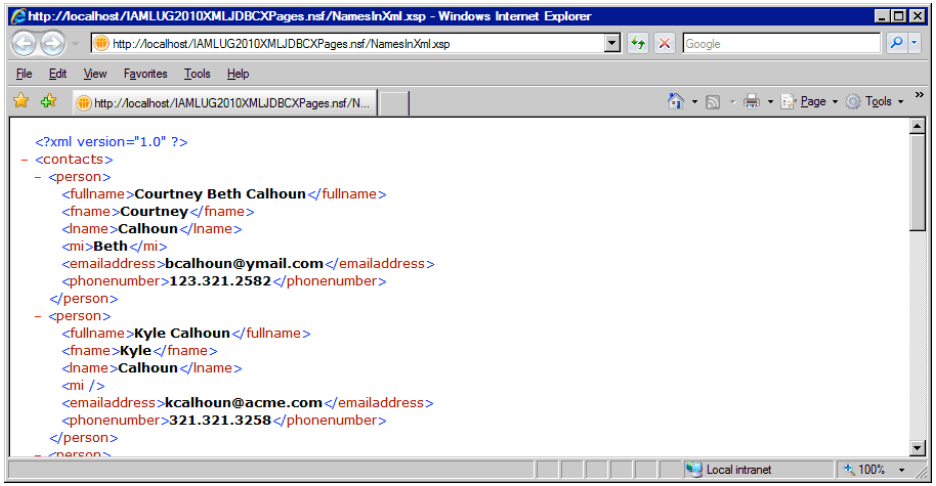


XML via an XAgent

- **XAgent Template Code**

```
// The Faces Context global object provides access to the servlet
environment via the external content
var extCont = facesContext.getExternalContext();

//Faces Context Response Writer is similar to the lotuscript print statement
// and the Java printwriter object to produce output to the browser.
// There is also a getResposneStream() for producing binary output.
var pageOutput = facesContext.getResponseWriter();

// The servlet's response object provides control to the response object
var pageResponse = extCont.getResponse();

// The content type hast to be set to the output MIME type.
// For Example text/html, text/xml, application/json
pageResponse.setContentType("text/xml");
// Additional methods of the response object can be set.  The following
keeps the content from being cached.
pageResponse.setHeader("Cache-Control", "no-cache");

<< use pageOutput.write();  to write content to the browser >>

// Tell the browser the content is complete
pageOutput.endDocument();
// Close the output stream
pageOutput.close();
//  Terminates the request processing lifecycle.
facesContext.responseComplete();
```

XML output in the Browser



Why XAgents and not regular agents?

- Traditional LotusScript and Java Agents execution requires the code to
  - Be read from the .nsf (disk or cache)
  - Placed in Memory
  - Executed
  - Taken out of memory
- This happens EVERY TIME the agent is called

Why XAgents and not regular agents?

- XAgents are
  - Compiled in to Servlets at runtime
  - Placed in memory the first time they are executed
  - Stay in memory
    - Until removed to accommodate other resources
    - XPage application engine is shut down

Demo

What We'll Cover …

- Producing XML from Domino
- Transforming XML as XPage Content
- Parsing XML as XPage Content
- RDBMS Data in an XPage using ZERO JAVA CODE !!!
- RDBMS Data in an XPage using JDBC
- Wrap-up/Summary/Questions

Transforming XML as XPage Content

- There are many instances when the XML data does not need to be manipulated, simply displayed
- The easiest way to do that is transform the XML via XSLT
- Transforming XML to HTML via XSLT allows transformed content to be written directly to the page

Transforming XML

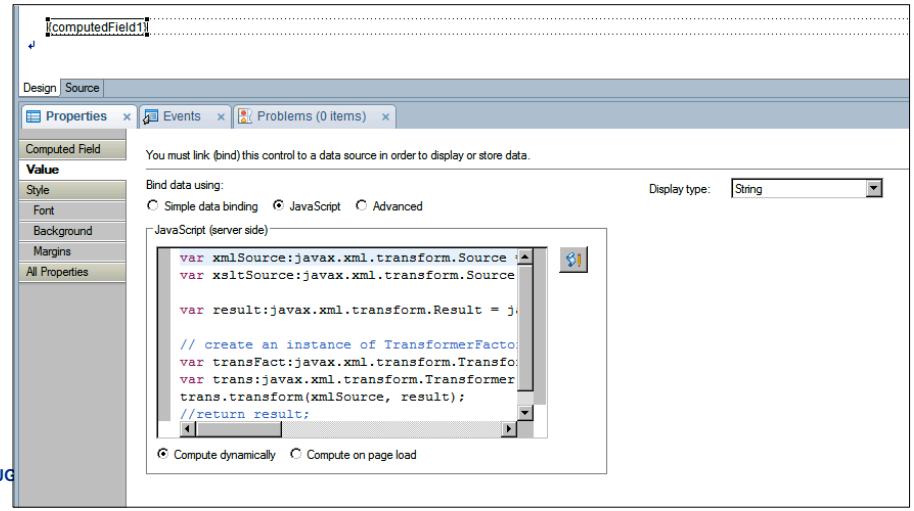
- The process for transforming XML into something else via XSLT is always the same
  - Get the XML
  - Get the XSLT
  - Perform the transformation and return the results
  - Use the results

Transforming Via Server Side JavaScript (SSJS)

- Server Side JavaScript has access to the complete Java core api
  - This includes xml parsing and xslt processing apis
    - javax.xml.*
- This solution takes the downstream browser out of the equation
  - The transformation happens on the server
  - Results are displayed in specified control

Transforming Via Server Side JavaScript (SSJS)

- Add a computed field control to the XPage where the transformed results will be displayed
- Add the code in the Value section of the Properties tab



Transforming Via Server Side JavaScript (SSJS)

- Create the input sources by pointing to the URL's for the XML and the Stylesheet
- Create a result variable to hold the transformed results
- Create the transformation factory and the transformer engine variables
- Perform the transformation
- Return the results

```
var xmlSource:javax.xml.transform.Source =
javax.xml.transform.stream.StreamSource("http://localhost/IAMLUG2010XMLJDBCXPages.nsf/NamesInXml.xsp");

var xsltSource:javax.xml.transform.Source =
 javax.xml.transform.stream.StreamSource("http://localhost/IAMLUG2010XMLJDBCXPages.nsf/names.xsl");

var result:javax.xml.transform.Result = javax.xml.transform.stream.StreamResult(facesContext.getResponseWriter());

// create an instance of TransformerFactory
var transFact:javax.xml.transform.TransformerFactory = javax.xml.transform.TransformerFactory.newInstance( );
var trans:javax.xml.transform.Transformer = transFact.newTransformer(xsltSource);
result = trans.transform(xmlSource, result);
return result;
```

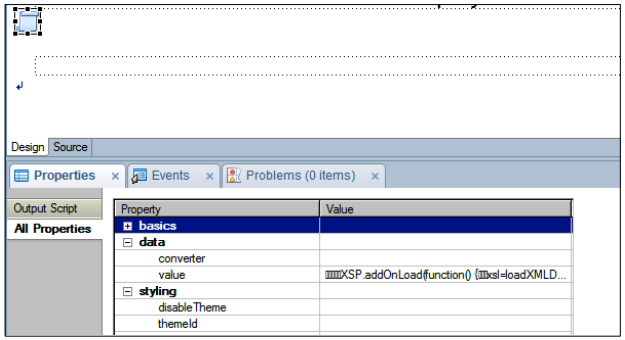
Demo

Transforming Via Client Side JavaScript (CSJS)

- **Transforming XML via Client Side JavaScript is a little less straight forward**
  - **Your code has to take browser type into account**

Transforming Via Client Side JavaScript (CSJS)

- Add a Panel (or div or span) control to display the results
- Add a Script Output control to you XPage
  - Add the CSJS to the value property under the data section of the All Properties section of the Script Output Properties tab
    - GOTCHA: After code has been added you are unable to get back to the Script Editor window.
    - FIX: Edit Code by switching to the source view of the XPage



Transforming Via Client Side JavaScript (CSJS)

- The process is the same as for SSJS
  - Get the XML and Stylesheet
  - Test for browser type
    - Call the browser appropriate transformation method
  - Write the results to the named control

```
XSP.addOnLoad(function() {

xsl=loadXMLDoc("http://localhost/IAMLUG2010XMLJDBCXPages.nsf/names.xsl");

xml=loadXMLDoc("http://localhost/IAMLUG2010XMLJDBCXPages.nsf/NamesInXml.xsp");
// code for IE
if (window.ActiveXObject)
  {
  ex=xml.transformNode(xsl);
  XSP.getElementById("#{id:displayResults}").innerHTML=ex;
  }
// code for Mozilla, Firefox, Opera, etc.
else if (document.implementation && document.implementation.createDocument)
  {
  xsltProcessor=new XSLTProcessor();
  xsltProcessor.importStylesheet(xsl);
  resultDocument = xsltProcessor.transformToFragment(xml,document);
  XSP.getElementById("#{id:displayResults}").appendChild(resultDocument);
  } });
```

Demo

Transforming RSS Feeds

- Another practical example of transforming XML to HTML on the fly is in the rendering of RSS News feeds
- RSS is a standard XML format
  - This means the same stylesheet can be applied to “n” number of RSS feeds

Demo

What We'll Cover …

- Producing XML from Domino
- Transforming XML as XPage Content
- Parsing XML as XPage Content
- RDBMS Data in an XPage using ZERO JAVA CODE !!!
- RDBMS Data in an XPage using JDBC
- Wrap-up/Summary/Questions

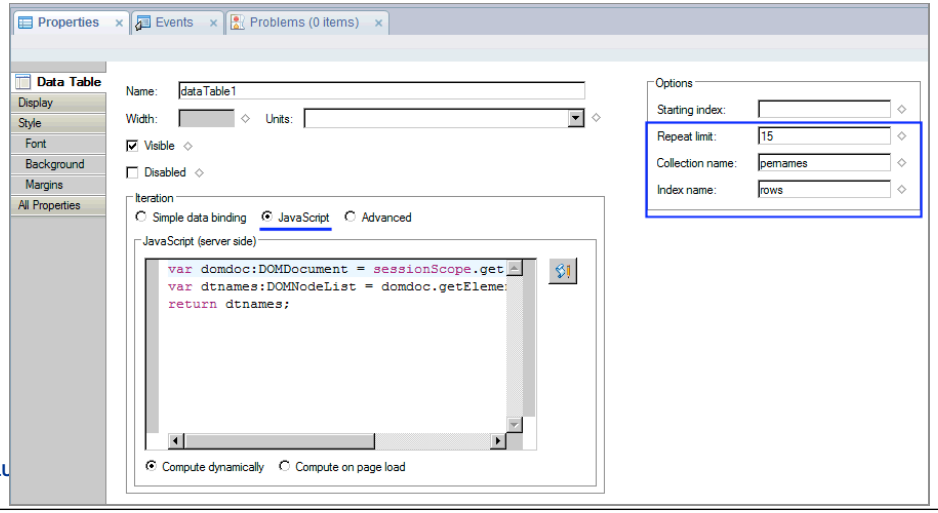
Parsing XML as XPage Content

- If the XML needs to be processed in some way that XSLT can not handle then an alternative is to parse the XML and use native XPage controls to produce the output
- Create a new XPage
- Add Code to the XPages afterPageLoad event to read and parse the XML
- Store the parsed XML in a session scope variable

```
//Create the XML Document
var parsedxml:org.w3c.dom.Document = null;
//Create the Parser Factory and document builder
var domfactory:javax.xml.parsers.DocumentBuilderFactory  =
javax.xml.parsers.DocumentBuilderFactory.newInstance();
var xmldocument:javax.xml.parsers.DocumentBuilder  = domfactory.newDocumentBuilder();

//Read the XML from the XAgent
var parsedxml = xmldocument.parse("http://localhost/IAMLUG2010XMLJDBCXPages.nsf/NamesInXml.xsp");


//Store the parsed XML to a session scope variable.
sessionScope.put("names",parsedxml);
```

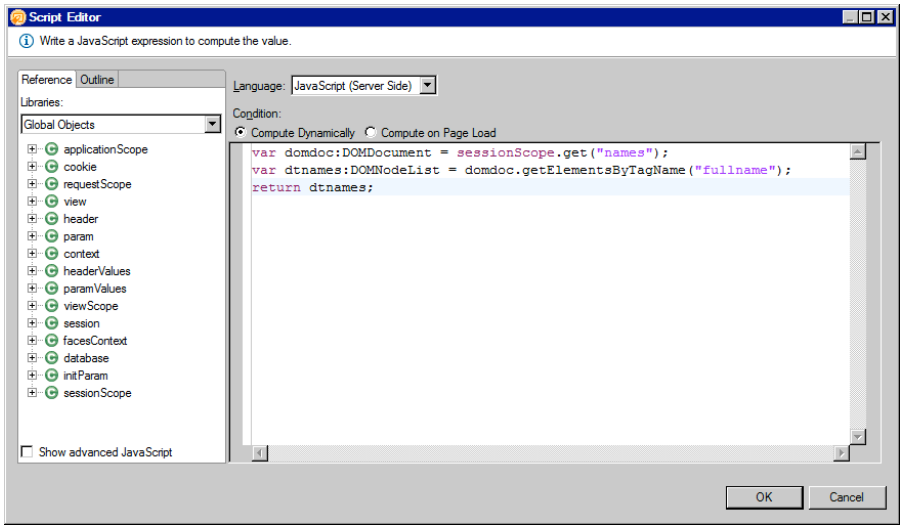
Parsing XML as XPage Content

- Add a Data Table control to the XPage
- Set the Repeat Limit, Collection name and Index name



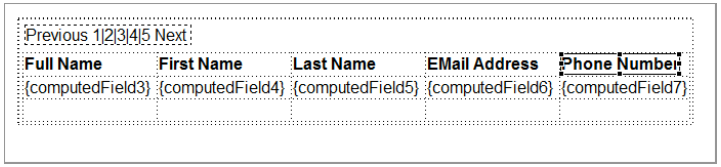
Parsing XML as XPage Content

- Set the Iteration to “JavaScript” and add the code to read the XML data



Parsing XML as XPage Content

- Add as many columns as is needed to display the data from the XML
- Add a Header using a Label control
- Add a computed field to compute the value from the XML



Parsing XML as XPage Content

- In the computed field add code to read the parsed XML variable from the scope variable
  - Create a DOM Node list from the parsed XML of the values to be displayed in this column
  - Create a DOM Node from the list using the index of the Data Table
  - Get that node value and return it to the column

```
//Read the Parsed XML variable stored in the session scope variable
var domdoc:DOMDocument = sessionScope.get("names");
//Create a node list out of the specific XML element to be displayed
//in this column
var dtnames:DOMNodeList = domdoc.getElementsByTagName("fullname");
//From the node list create a DOMNode of the specific element using the
//rows index variable from the data table
var nnode:DOMNode = dtnames.item(rows);
//Return the value of that node to the column
return nnode.getFirstChild().getNodeValue();
```

Demo

What We'll Cover …

- Producing XML from Domino
- Transforming XML as XPage Content
- Parsing XML as XPage Content
- RDBMS Data in an XPage using ZERO JAVA CODE !!!
- RDBMS Data in an XPage using JDBC
- Wrap-up/Summary/Questions

RDBMS Data in XPages

- Domino has the ability to integrate with Relational Database Management Systems (RDBMS) via included and third party tools, for example;
  - DCRs
  - DECS
  - LEI
  - Third Party tools
  - JDBC
- These tools can be used to work with RDBMS data in XPages

Adding RDBMS data to an XPage using DECS

- DECS which is included with the Domino Server has the ability to connect to the most common RDBMS including
  - DB2®
  - ODBC
  - Oracle®
  - OLE DB
  - Sybase®
- No additional third party tools or drivers are necessary in order to connect to the back end systems.
- This solution is Domino Server based as it requires the Domino Servers DECS task

Adding RDBMS data to an XPage using DECS

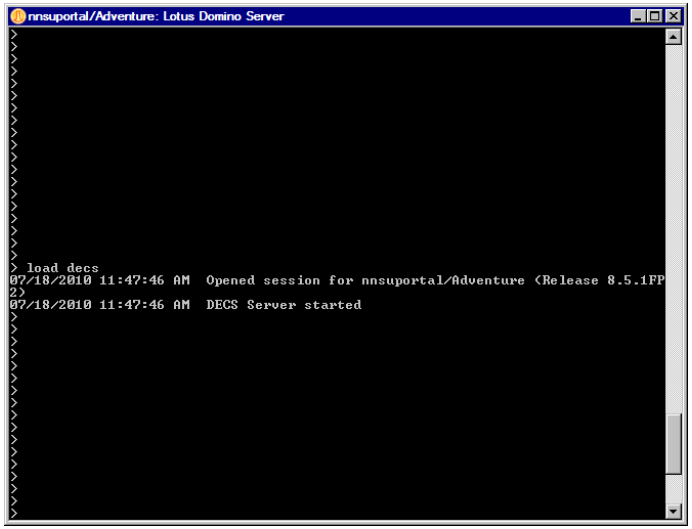
- Use the following steps to add RDBMS data to an XPage via DECS
  - Run the Domino Servers DECS task
  - Create a Domino Form
    - To be used in a DECS Virtual Field Activity
    - Mapped to the RDBMS source via DCR's
  - Map the Domino Form fields to the RDBMS system
    - Via a DECS Virtual Field Activity
    - Via DCR's
  - Create a view that displays
    - Key field Data
    - Locally stored data

Adding RDBMS data to an XPage using DECS

- **Use the following steps to add RDBMS data to an XPage via DECS (cont)**
  - **Create an XPage to view the Key/local field data and provide links to the documents (A View XPage)**
  - **Create an XPage that uses the DECS/DCR form as a data source**

Adding RDBMS data to an XPage using DECS

- Load the DECS Domino Server task
  - To have the DECS task load at server start up add "DECS" to the Servers notes.ini "ServerTasks=" entry.



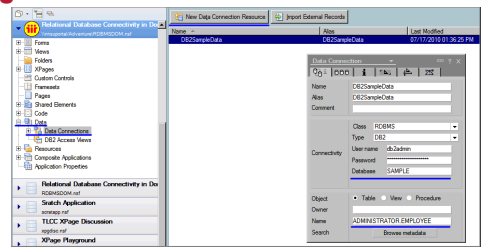
Adding RDBMS data to an XPage using DECS

- Create a Domino Form to be used in the DECS virtual field activity or to be mapped via DCRs
- The following form contains the fields that map to the SAMPLE database schema on the DB2 server

Employee Number:	EMPNO
First Name:	FirstName
Middle Initial:	MidInitial
Last Name:	LastName
Department:	Department
Phone Number:	PhonNo
Hire Date:	HireDate
Position:	Position
Education:	Education
Sex:	Sex
Birth Date:	BirthDate
Salary:	Salary
Bonus:	Bonus
Commission:	Commission

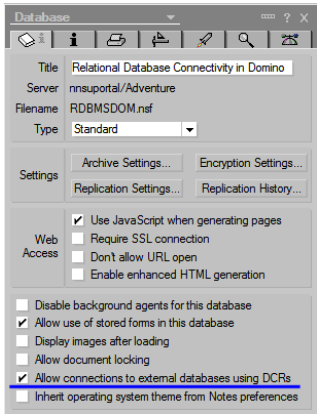
Adding RDBMS data to an XPage using DECS

- In this example we will be using a DCR to connect to the RDBMS system
  - In Domino Designer expand the Data section
    - Double click on "Data Connections"
    - Click the "New Data Connection Resource" action in the action bar
    - Provide the information to connect to the RDBMS system
    - In this example we are connecting to a DB2 system and accessing the EMPLOYEE table in the SAMPLE database.



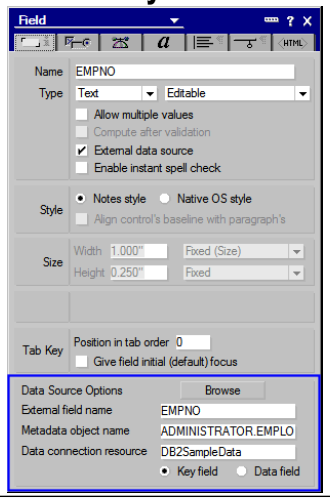
Adding RDBMS data to an XPage using DECS

- Enable the application property to allow connections to external databases using DCR's



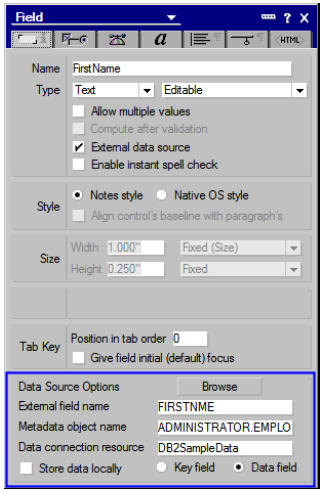
Adding RDBMS data to an XPage using DECS

- Map the fields in the previously created form to the RDBMS fields using the DCR
  - At least one field has to be a "Key Field"
  - Key field data is automatically stored with the .nsf.



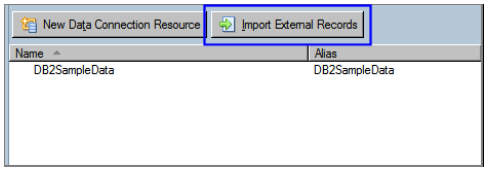
Adding RDBMS data to an XPage using DECS

- **All other fields are mapped as "Data Fields"**
  - By default data field values are NOT stored with the .nsf
  - In order for data to be used in Domino Views the "Store data locally" check box must be selected.



Adding RDBMS data to an XPage using DECS

- If data already exists in the RDBMS system then optionally import those records to Domino
  - Disable the application property "Allow connections to external databases using DCRs"
  - In Domino Designer Data | Data Connections, select the data connection to be used
    - Click on the "Import External Records" action in the action bar
  - Re-Enable the application property "Allow connections to external databases using DCRs".



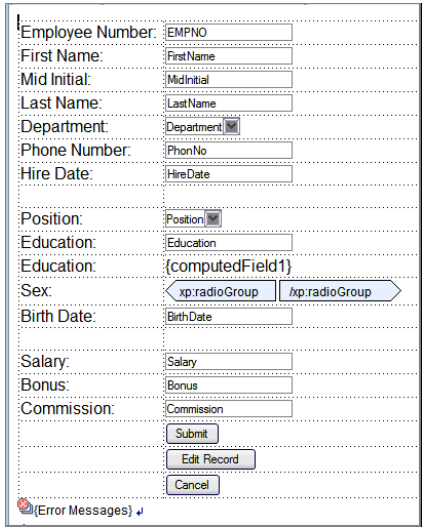
Adding RDBMS data to an XPage using DECS

- **Create a Domino View to display the Key field data and any locally stored data**
  - **In the following example the Employee Number is the Key Field**
  - **The First and Last Names are stored with the .nsf.**

Employee Number	First Name	Last Name
000010	CHRISTINE	HAAS
000020	MICHAEL	THOMPSON
000030	SALLY	KWAN
000050	JOHN	GEYER
000060	IRVING	STERN
000070	EVA	PULASKI
000090	EILEEN	HENDERSON
000100	THEODORE	SPENSER
000110	VINCENZO	LUCCHESSI
000120	SEAN	O'CONNELL
000130	DELORES	QUINTANA
000140	HEATHER	NICHOLLS
000150	BRUCE	ADAMSON
000160	ELIZABETH	PIANKA
000170	MASATOSHI	YOSHIMURA
000180	MARILYN	SCOUTTEN
000190	JAMES	WALKER
000200	DAVID	BROWN
000210	WILLIAM	JONES
000220	JENNIFER	LUTZ
000230	JAMES	JEFFERSON
000240	SALVATORE	MARINO

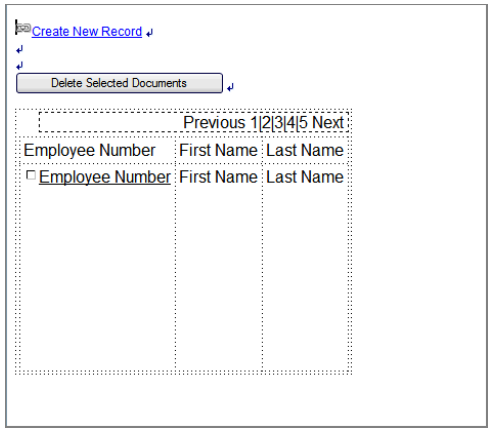
Adding RDBMS data to an XPage using DECS

- **Create an XPage to display the Documents**
  - Include buttons for
  - Submitting
  - Switching to Edit mode
  - Canceling back to the View XPage



Adding RDBMS data to an XPage using DECS

- Create an XPage to display the View
  - Add links/buttons to
    - Create a new record
    - Delete selected documents
  - Enable one of the columns to Link to the documents.



Adding RDBMS data to an XPage using DECS

- When the XPage View is displayed
  - All of the records with key and locally stored data will be displayed
- When the view link is clicked
  - The DCR form mapping will be used to retrieve the rest of the field data from the back end RDBMS system and display it
- The use of DECS/DCRS is completely transparent to the end user
- The Developer did not write a SINGLE LINE OF JAVA !!
  - Or much of any code for that matter

Demo

What We'll Cover …

- Producing XML from Domino
- Transforming XML as XPage Content
- Parsing XML as XPage Content
- RDBMS Data in an XPage using ZERO JAVA CODE !!!
- RDBMS Data in an XPage using JDBC
- Wrap-up/Summary/Questions

RDBMS Data in an XPage using JDBC

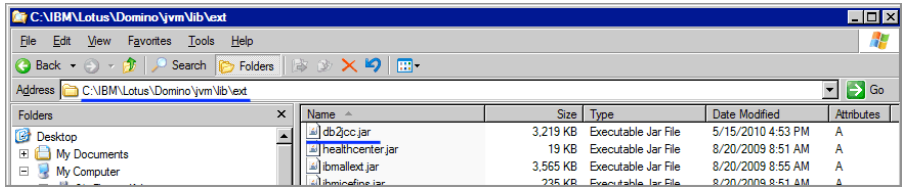
- JDBC is an industry standard architecture for accessing RDBMS data using Java Code
- The majority of the major RDBMS vendors provide JDBC drivers for accessing their systems
- Data from the RDBMS can be rendered in the XPage using both CSJS as well as SSJS

RDBMS Data in an XPage using JDBC and CSJS

- To connect to a RDBMS via JDBC using Client Side JavaScript developers will need to complete the following steps
  - Acquire the vendor appropriate JDCB driver library for accessing the RDBMS
    - Deploy the Library file for development and production
  - Write the Java Agent* code needed to make the connection to the RDBMS and read the data
    - Call this Java code from an XPage Client Side JavaScript event

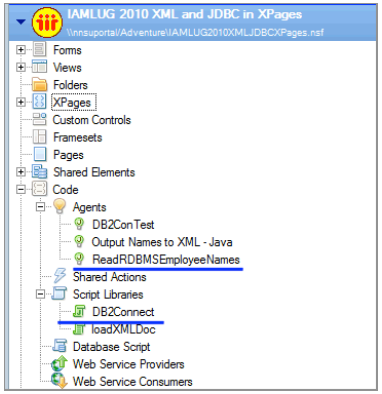
RDBMS Data in an XPage using JDBC and CSJS

- Acquire the JDBC driver libraries
  - These files are available from the RDBMS vendors web site
  - Also usually installed with the RDBMS server
  - Consult with your DB Administrator as to the best JDBC driver library to be using in your environment
- Deploy the JDBC libraries to the appropriate directories
  - It is a best practice to make the library files available to the entire Notes/Domino system
    - Copy the .jar files to
      - *Notes/Domino <installation Directory>/jvm/lib/ext folder*



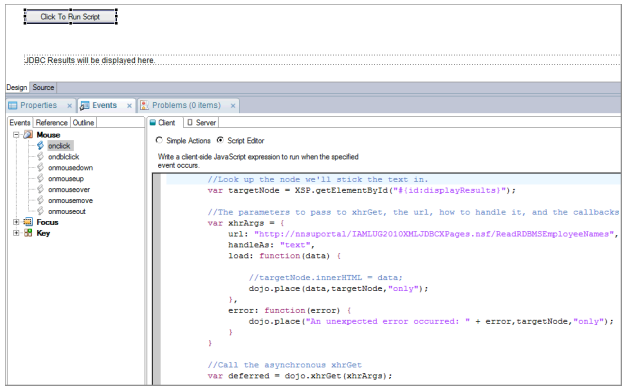
RDBMS Data in an XPage using JDBC and CSJS

- Create a Java Agent that will use the JDBC drivers to connect to the RDBMS and return the data
  - In this case we will be using
    - Java Agent and a Java Script Library



RDBMS Data in an XPage using JDBC and CSJS

- Use Dojo AJAX calls to retrieve the data from the Java Agent
  - In the Client side onclick event of a button control
    - Add Java Script that will
      - *Make the AJAX call and return the data*
      - *Output the returned data to a panel on the XPage*



Demo

RDBMS Data in an XPage using JDBC and SSJS

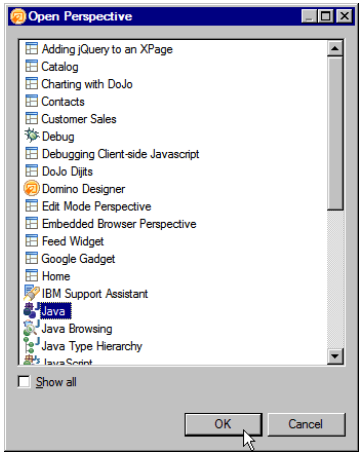
- In order to connect to a RDBMS via JDBC using SSJS developers will need to complete the following steps
  - Acquire the vendor appropriate JDCB driver library for access the RDBMS
    - Deploy the Library file for development and production
  - Write the Java code needed to make the connection to the RDBMS and manipulate the data.
    - Call this Java code from an XPage Server Side JavaScript event
- NOTE: XPages do NOT run in the JVM as Domino Web Pages or Notes Agents

RDBMS Data in an XPage using JDBC and SSJS

- Acquire the JDBC driver libraries
  - Same as for CSJS
- Deploy the JDBC libraries to the appropriate directories
  - For XPages to access the JDBC drivers the jar file should be deployed to
    - Notes/Domino <installDirectory>\xsp\shared\lib folder

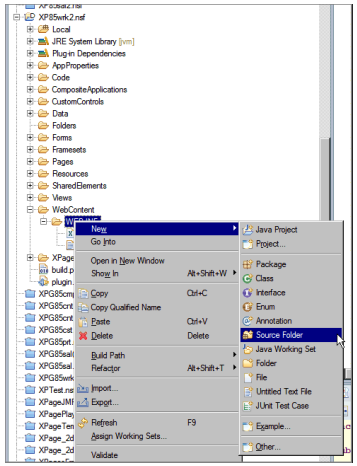
RDBMS Data in an XPage using JDBC and SSJS

- **Create the Java source code that is needed to access the RDBS system**
  - This code is NOT Java Agents but Java Class files created in the Java Perspective in DDE
  - Switch to the Java Perspective
    - From the menu choose Window | Open Perspective | Other...
    - Select Java and click OK

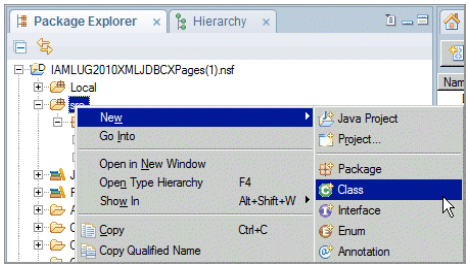


RDBMS Data in an XPage using JDBC and SSJS

- In the Package Explorer tab
  - Expand the project that will contain the Java code
  - Expand the WebContent | WEB-INF folder
    - Right click on the WEB-INF folder and select New | Source Folder from the context menu.

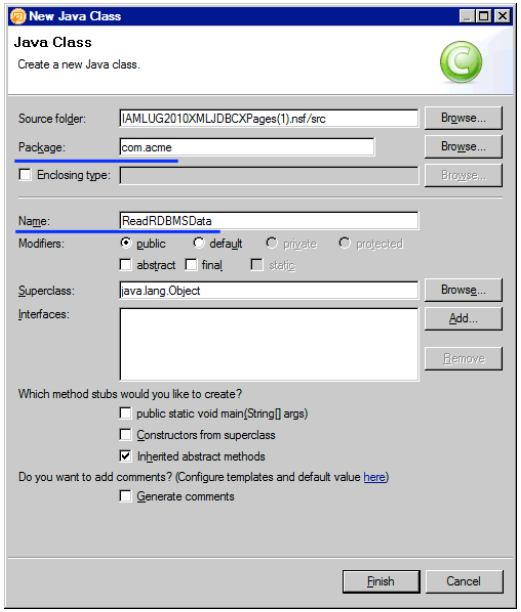


- The new source folder will appear at the top of the project
  - Right click on the source folder
    - Select New | Class from the context menu.



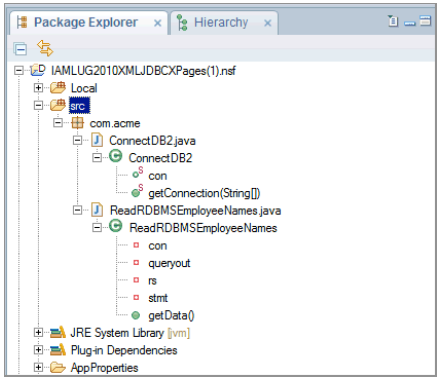
RDBMS Data in an XPage using JDBC and SSJS

- In the New Java Class dialog box
  - Provide
    - Package name
    - Class Name



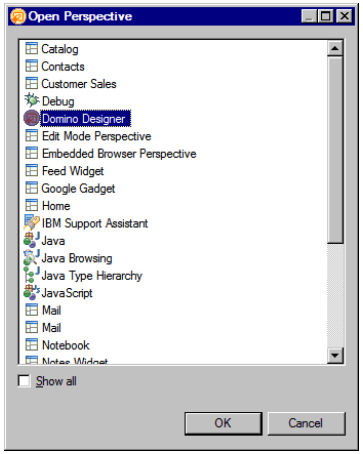
RDBMS Data in an XPage using JDBC and SSJS

- **Write the Java Code that will**
  - Connect to the RDBMS system
  - Read the data to be displayed on the XPage



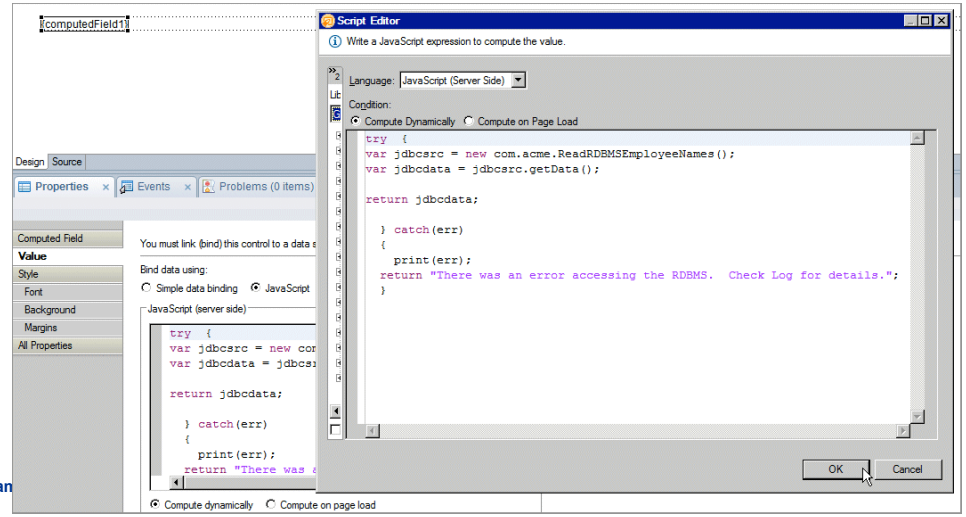
RDBMS Data in an XPage using JDBC and SSJS

- **Switch back to the Domino Perspective**
  - **From the Menu choose**
    - **Window | Open Perspective | Other**
    - **In the Open Perspective dialog box that opens select**
      - ***"Domino Designer" and click "OK"***



RDBMS Data in an XPage using JDBC

- **In a Server Side JavaScript event**
  - Add the code that will instantiate an instance of the Java class
  - Call the method that returns the data



Demo

What We'll Cover …

- Producing XML from Domino
- Transforming XML as XPage Content
- Parsing XML as XPage Content
- RDBMS Data in an XPage using ZERO JAVA CODE !!!
- RDBMS Data in an XPage using JDBC
- Wrap-up/Summary/Questions

Wrap Up

- XML data can be used on an XPage by
  - Transforming XML into HTML via XSLT
  - Parsing XML and writing the output to the page via
    - Client Side JavaScript
    - Server Side JavaScript
- JDBC data can be used on an XPage by
  - Using DECs, and ZERO Java Code
  - JDBC via
    - Client Side JavaScript
    - Server Side JavaScript

Resources

- Me
  - Consulting, Training, Mentoring
  - http://www.nnsu.com
    - Download presentations, code and tools
- TLCC - The Learning Continuum Company
  - http://www.tlcc.com
    - XPage, Java, XML and Web Services Curriculum
- XPages Blog
  - http://www.xpagesblog.com
    - Nothing but XPages tips, tricks and techniques

Resources

- IBM
  - DB2C
  - Developerworks
  - Designer Wiki's
- People you should follow
  - Matt White - XPages 101 Training
    - http://xpages101.net/
  - David Leedy- Notes in 9 - 9 min XPage videos
    - http://www.notesin9.com/
  - Mark Hughes - iPhone XPage Template
    - http://dominoextnd.blogspot.com/
  - Paul Withers - XPage blog
    - http://hermes.intec.co.uk/intec/blog.nsf

Q and A

If you send me EMail
Make sure the subject line contains
“View” somewhere in the subject line
or I will never see it

How to contact me:
Paul T. Calhoun
pcalhoun@nnsu.com