JavaBean/Backing Bean

Fetch NSF User Access/ACL, Hide/Show Controls

Xpages

System Requirements:
Download Domino Designer 8.5.2 Environment (DDE)
http://www.ibm.com/developerworks/downloads/ls/dominodesigner/

Introduction:
Use prior Login JavaBean and Xpages samples as guide, build a connection to a Lotus Notes Domino Database located on C Drive using previous information (LoginBean.SQL.java, xpnavlinks.xsp). Supply the Xpage the logged-in system/user ACL and other security access information. Hide/show link control(s) on the Xpages form at will...

Disclaimer:
Information contained in the following is presented as is. This tutorial assumes you have basic programming knowledge. All tutorials are based on an Eclipse/Eclipse-based software. Should you need to familiarize yourself with a certain Eclipse environment, prior to continuing this tutorial, please stop now and see our Tutorials page...

Copy and Paste Xpages link control Code
At this point we assume Domino Designer 8.5.2 is downloaded/installed, you've looked at the LoginBeanSQL class, and you added a link control to your xpnavlinks.xsp. Copy and paste below code samples to your environment, areas of interest have been highlighted for your convenience.

xpnavlinks.xsp

```
<xp:link escape="true" id="link1"
rendered="#{javascript:LoginBeanSQL.ACLAccessOkay()}" text="ACL Access"
value="/xpaclaccesscheck.xsp" style="color:rgb(0,128,128)"
target="colours"></xp:link>

<xp:link escape="true" id="link1"
rendered="#{javascript:LoginBeanSQL.SecurityAccessOkay()}" text="Security
Access" value="/xpsecurityaccesscheck.xsp" style="color:rgb(0,128,128)"
target="colours"></xp:link>
```

Copy and Paste Boolean Methods
Call below methods from LoginBeanSQL.java class, add these Boolean methods as new methods within the LoginBeanSQL.java class. You can also write a new class to do this and include the methods there. Copy and paste below code samples to your environment, reference highlighted portions.

ACLAccessOkay(); | SecurityAccessOkay();

declarations, getters and setters...

```
	//declare boolean variables for searching
	private boolean ACLAccessOkay;
	private boolean SecurityAccessOkay;
	//declare acl level variable, User attributes
	private String UserAttr;

/**
	 * @return the aCLAccessOkay
	 */
	public boolean isACLAccessOkay() {
		return ACLAccessOkay;
	}
	/**
	 * @param accessOkay the aCLAccessOkay to set
	 */
	public void setACLAccessOkay(boolean accessOkay) {
		ACLAccessOkay = accessOkay;
	}

	/**
	 * @return the securityAccessOkay
	 */
	public boolean isSecurityAccessOkay() {
		return SecurityAccessOkay;
	}
	/**
	 * @param securityAccessOkay the securityAccessOkay to set
	 */
	public void setSecurityAccessOkay(boolean securityAccessOkay) {
		SecurityAccessOkay = securityAccessOkay;
	}


	/**
	 * @return the userAttr
	 */
	public String getUserAttr() {
		return UserAttr;
	}
	/**
	 * @param userAttr the userAttr to set
	 */
	public void setUserAttr(String userAttr) {
		UserAttr = userAttr;
	}

/**

	 * Method: ACLAccessOkay()
	 * Created from Copy: 2012.08.26.4.14.AM
	 * ACL Access Boolean for Xpages Controls (Hide/Show)
	 *
	 */
```

```
// links control, items on form vanish if Security access is 0
// for ACL LEvel access to in NSF back-end
public boolean ACLAccessOkay() {
      //TO DO: Give Anonymous ACL Level Manager for this to work
      //you can set this up under Advanced Tab in Access Control
      //2012.08.26.1.55.AM
      try {
      //get the current database being used
      Database database= (Database) FacesContext.getCurrentInstance()
      .getApplication().getVariableResolver()
      .resolveVariable(FacesContext.getCurrentInstance(), "database");
      //add debug messages
      System.out.println("Checking Database for ACL: "+ database);
      //...
      //initialize user ACL
      ACL UserAcl = database.getACL();
      //declare variables to hold ACL Level
      String AclLevel = null;
    //...
      int intLevel = UserAcl.getInternetLevel();
      //pick off user expected ACL
      //load to variable and compare
      if (intLevel < ACL.LEVEL_DESIGNER)
        UserAcl.setInternetLevel(intLevel + 1);
      UserAcl.save();
      switch(UserAcl.getInternetLevel()) {
        case ACL.LEVEL_NOACCESS:
            AclLevel = "no"; break;
        case ACL.LEVEL_DEPOSITOR:
            AclLevel = "depositor"; break;
        case ACL.LEVEL_READER:
            AclLevel = "reader"; break;
        case ACL.LEVEL_AUTHOR:
            AclLevel = "author"; break;
        case ACL.LEVEL_EDITOR:
            AclLevel = "editor"; break;
        case ACL.LEVEL_DESIGNER:
            AclLevel = "designer"; break;
        case ACL.LEVEL_MANAGER:
            AclLevel = "manager"; break; }
      //load results to variable
    //use in if statement, return true
      UserAttr= AclLevel.toString();
      System.out.println("User is a "+ UserAttr + " in this db...");
      //...
      }catch (NotesException e) {
            //print this error to the server
            e.printStackTrace();
      }
      //find out if the ACL is of Level manager
      //in which case user should be able to see/edit control(s)
      if(this.UserAttr !="manager")
      //DO NOT give user ability to update field if a NOT manager
      //Disable this field at once on the Xpage

      //TO DO: Play around with images at this point, load from URL
      //especially if using the link control, add an image
      //user can click images/icons for add/edit/delete docs through Xpages
            return true;
```

```
		//let user edit field at will, user is a manger,
		//this field should not be disabled
		return false;
	}
	/**

	 * Method: SecurityAccessOkay()
	 * Created: 2012.08.26.2.08.AM
	 * Security Access Boolean for Xpages Controls (Hide/Show)
	 *
	 */
	// links control, items on form vanish if Security access is 0
	// for viewaccess, updateaccess, addaccess, and deleteaccess in NSF back-end
	public boolean SecurityAccessOkay() {
		// let's add a try catch here, to grab errors near the end
		try {
			Document doc;
		//BEGIN DEBUG
			Database database = (Database) FacesContext.getCurrentInstance()
			.getApplication().getVariableResolver().resolveVariable(
					FacesContext.getCurrentInstance(), "database");
			System.out.println("Database Obtained..." + database);
		// Find the view in question
			View view = database.getView("SecurityAccessView");
			System.out.println("View Obtained..." + view);
		//initialize doc collection for searching
		//TO DO: Add adding code to read updateaccess, addaccess, and
deleteaccess in NSF back-end
			DocumentCollection dc =view.getAllDocumentsByKey("ViewAccess",
true);//
			System.out.println("Collection Obtained..." + dc);

			  dc = database.FTSearch("Yes");
		//Get first Document from current view
				doc = (Document) dc.getFirstDocument();
				System.out.println("All systems are go, looping..." + dc);
				// Create a while loop to process the document in the View
				while (dc!=null) {
					//add more code here, do stuff
					System.out.println("Results are in..." + dc);
					//Let user see field/control on Xpages
					return true;
				}
		} catch (Exception e) {
			e.printStackTrace();
		}
		//User should NOT see field/control on Xpages
		return false;
	}
```

Added information

Call methods from a button using below.

```
	//...
	//load ACL access level
	ACLAccessOkay();
	//load security access from security table
	SecurityAccessOkay();
```

Create Form/View to house secure documents

Secured documents should be created either on separate servers or the current. These documents should be restricted to Admin users ONLY. Create a form and a view with field names stated below. Set Datatype on all fields as Text.

FormName

SecurityAsccess

UserID
RoleID
ViewAccess
AddAccess
UpdateAccess
DeleteAccess

ViewName

SecurityAccessView

UserID
RoleID
ViewAccess
AddAccess
UpdateAccess
DeleteAccess

Screenshot

UserID	RoleID	ViewAccess	AddAccess	UpdateAccess	DeleteAccess	DateCreated
username101	Admin	Yes	**Yes**	Yes	Yes	08/18/2012 10:18:46 PM
username102	User	Yes	**Yes**	No	No	08/18/2012 10:14:02 PM

WARNING:

Restrict column visibility using formula code, put Anonymous in the role of Admin role through the ACL. Respectively, grant more explicit groups identical access when deploying to actual Notes Servers. We will skip additional measures on the subject for the time being, Anonymous should do just fine for our Local (http://localhost or http://127.0.0.1) Server.

Conclusion:

You can now retrieve ACL level access and other security access information collected/stored restricted in secured (Admin ONLY) documents; allowing show/hide controls on Xpages forms based on these level of access or security access permissions set forth.

Questions, comments, please post a brief message on our Contact form on the main site.

Thank you for coming...