NSF Document JDBC Provider

The NSF Document JDBC Provider, provides a way to store JDBC Connection information using Documents in an NSF, instead of within an xml file located within the source code.

This helps to avoid placing passwords within the design of the application, and allows them to be configurable at runtime / deployment.

You can set up information as a 'Local' provider, which means that the JDBC Configuration is read from the current database, or you can also provide it using a 'global' provider, which means that it will look for connection information in a predetermined NSF (server-wide)

1. Feedback / Suggestions / Troubleshooting

I would love to hear any feedback, you have!

If this worked great for you then just drop a 'worked great' comment somewhere.

If there were problems or the instructions were too vague please let me know so I can fix it up a bit

The best way to provide feedback will be through the Project's Issues Page[1]

2. Demonstration Video

Check out this demonstration video on youtube[2] of the JDBC Provider in action

3. Installation

To use the NSF Document JDBC Provider, make sure you have installed the ExtLibX plugins to your Domino Designer and Domino Server.

Then make sure you have selected the `com.ibm.xsp.extlibx` library in the XSP Properties file for that application. You will also need to select the normal Extension library, and it's relational library.

[1] https://github.com/camac/XPagesExtLibX/issues
[2] https://youtu.be/7sassOh-wkw

Then:

- Set up the JDBC Configuration Design Elements in the database that will hold the configuration
- Configure the application so it knows where to load JDBC Configuration from (Local, Global or Both)
- Set up your JDBC Configuration Documents in the database that you are using for configuration

3.1. Using a Local Provider

To use a Local Provider, put the following entry in your applications xsp.properties file

```
xsp.jdbc.nsfdocumentprovider.local=true
```

3.2. Using a Global Provider

To use a Global Provider, put the following entry in your application's xsp.properties file, where <filepath> is the file path of your NSF that has the JDBC Configuration. If you want to use the same JDBC Configuration database for all applications on the server, then you can supply this via the server's xsp.properties file

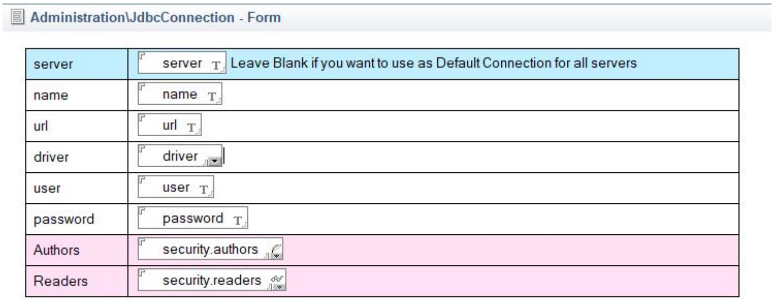
```
xsp.jdbc.nsfdocumentprovider.global.filepath=<filepath>
```

3.3. Setting up the JDBC Configuration Design Elements

The JDBC Provider expects a certain convention to discover the connection information. It will search a specific view, for a JDBC Configuration document, and when it finds the document it will then expect certain field to find the configuration information.

Configuration Document Form Design

The form that holds the JDBC Connection Configuration details is expected to have the following fields:



Administration\JdbcConnection - Form

server	server Leave Blank if you want to use as Default Connection for all servers
name	name
url	url
driver	driver
user	user
password	password
Authors	security.authors
Readers	security.readers

server

This field can be left blank if the JDBC Connection information is valid for any Domino server. However sometimes you may have different configuration for different locations. e.g. ServerA may access an SQL Server within the same office and ServerB may access a similar SQL Server that is located in it's own office. These SQL Servers may have different URLs and even different username and passwords. These 2 configurations can be stored as 2 documents but using the same connection name. The JDBC Provider will use the configuration that applies to that domino server

name

This is the connection name of the JDBC Connection. This will be used within XPages or Java code to obtain the connection

url

This is the connection url used to connect to the JDBC connection

user

This is the username used for the JDBC connection

password

This is the password used for the JDBC connection

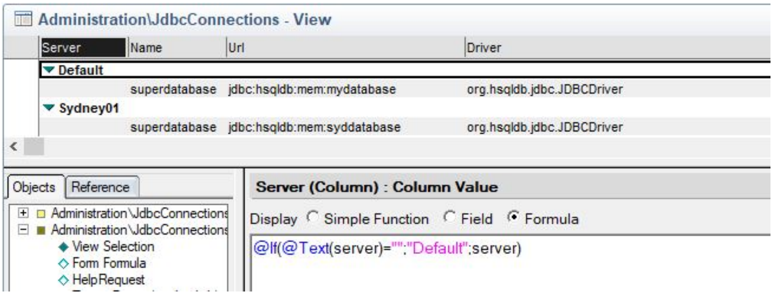
Optional / Recommended

You can also add an authors field and a readers field and restrict Read access to Administrators and also the XPages Database Signer.

This way, the connection password is kept secure from prying eyes.

View Design

The JDBC Connection provider expects to find a view named **JDBCConnections**



Administration\JdbcConnections - View

Server	Name	Url	Driver
Default			
	superdatabase	jdbc:hsqldb:mem:mydatabase	org.hsqldb.jdbc.JDBCDriver
Sydney01			
	superdatabase	jdbc:hsqldb:mem:syddatabase	org.hsqldb.jdbc.JDBCDriver

Objects
Reference
Administration\JdbcConnections
Administration\JdbcConnections
View Selection
Form Formula
HelpRequest

Server (Column) : Column Value

Display ○ Simple Function ○ Field ● Formula

@If(@Text(server)="";"Default";server)

The first column should be categorised, and use the following formula:

```
@If(@Text(server)="";"Default";server)
```

This will group all the configurations that are not server-specific under the one 'Default' Category.

The second column should be the **name** field and sorted (so it can be looked up). After that you can put whatever columns you like

Basically, the provider will do a lookup using Current ServerName + Connection Name. If it doesn’t find an entry for that, it will then try 'Default' + Connection Name

3.4. Troubleshooting / Debugging

If things don’t seem to be working, try turning on the Relational Logging so you can see what is going on on the Domino Console (or in the log files in IBM_TECHNICAL_SUPPORT folder of the Domino Data directory.

To turn the logging on, put the following entry into the rcp **<dominodata>/domino/workspace/.config/rcpinstall.properties**

```
com.ibm.xsp.extlib.relational.level=FINEST
```

Note: The initialisation stage is probably the most important to see what is going on, and this only happens when you first fire up an application, so maybe you need to restart the server

4. Ideas for Improvements

- Include the extra Connection Pool parameters in the Configuration Document (e.g. max connections etc)
- Add a menu option in Domino Designer to automatically install the JDBC Configuration Elements