

# IBISServer

Version 0.1

## Design Concepts User Notes

Jack Park

Latest: 20090731

# Big Picture

- Server to provide online persistence for IBIS documents
  - Options
    - Simple flat-file persistence with indexing
      - Add full text search
      - Separate by IBIS doc type (e.g. Compendium, bCisive, ...)
    - Database persistence with indexing
      - Add full text search
      - Separate by IBIS doc type
    - Either of the above plus conversion to a common format

# First Plan

- Simple Flat-file persistence with file indexing
  - Use cases
    - Get a list of document names
      - Include document type
    - Get a named document
    - Put a document
    - Remove a document
  - Protocol
    - Requires authentication
    - REST API to indicate document type
  - Format
    - It's flat file with no interpretation
      - Can be XML, JSON, or other text format

# REST Format

- Basic Get (HTTP Get)
  - `http://<server>/ws/get/<platform>/<filename>`
    - Response is an XML string
- Basic Put (HTTP Post) requires authentication
  - `http://<server>/ws/put?filename=...`
- Basic Remove (HTTP Post) requires authentication
  - `http://<server>/ws/remove?filename=...`
- Get Index (HTTP Get) (list files)
  - `http://<server>/ws/`
    - Response is a JSON string ,e.g.:
      - `[[ggw-3.xml, cm, Grains Gone Wild, jackpark], [TrivialMap.xml, cm, Trivial Map, jackpark]]`

# Current Design Issues

- Should we require authentication for all actions?
  - Inclined to enforce authentication for the time being
    - Could return to authentication only on post operations
    - Presently require authentication to post.
    - Administrative authentication built into config file for a default administrator, used for adding users
- XML serialization (compare to JSON)?
  - Will stay with Common XML Format for the time being.
  - JSON later

# First Implementation

- Eclipse Jetty project
  - Model
    - IBISModel
  - View
    - IBISViewHandler
  - Control
    - IBISServlet

# First Document (Platform) Types

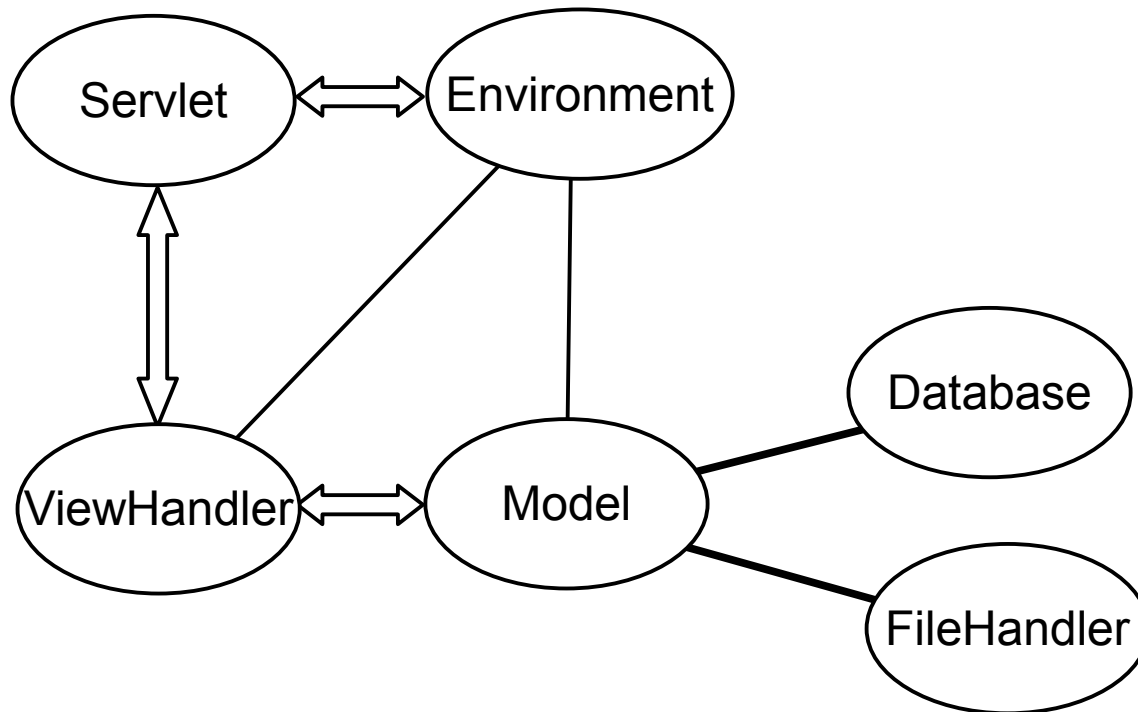
- Defined in a configuration file
  - Compendium
    - “cm”
  - DebateGraph
    - “dg”
  - Deliberatorium
    - “dm”
  - Cohere
    - “co”

# Extensions

- Shell code but not implemented in Version 0.1
- Conversation Parser
  - Parses IBIS conversations into a rooted tree made of `IBISNode` objects
- Plugin Agents
  - Full text indexing of all `IBISNode` objects
  - Clustering of *child nodes*
    - Returns a list of named clusters for the child nodes of the same *node type*
      - e.g. Cluster all Position nodes that are children of an Issue node
      -



# General Architecture



Pattern: Model-View-Controller

# Components

- Servlet
  - Creates Environment
  - Creates ViewHandler
- Environment
  - Creates other components
  - Provides utility functions
- ViewHandler
  - Creates Model
  - Provides responses to Servlet queries
- Model
  - Processes data
  - Creates and owns Database and FileHandler

# Database

- User Data (Derby)
  - Authentication of users
  - Future storage of IBISNode objects
- File Data
  - Maintains a list of
    - File paths
    - Platforms (e.g. Compendium, DebateGraph,...)
    - File descriptions
    - File authors

# FileHandler

- Handles persistence of flat file objects
  - IBIS conversations in XML format
    - Common Format

# Conversation Extension

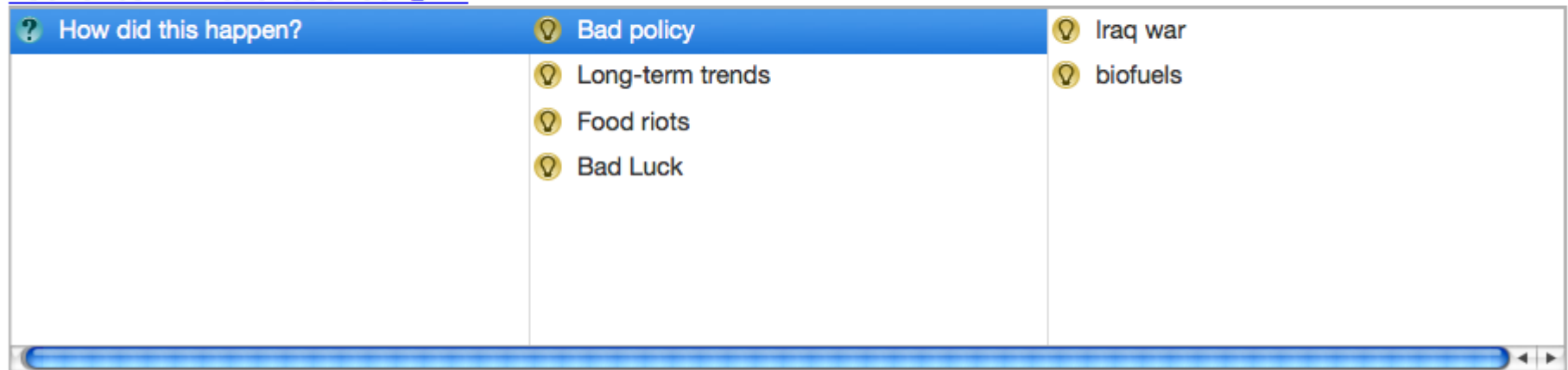
- Framework for extending IBIS Server capabilities and services
- XML parser
  - Read IBIS XML file
  - Create a rooted tree of `IBISNode` objects
    - Tree can be passed to plug-in agents for further processing

# Conversation Viewing—Future



## Available Dialogue Maps

[13010765211207678551684\\_DM](#)



## position Node

**Id:** 13010765211207678702901

## Subject

Bad policy

## Details

Image adapted from a different project that will share code with IBISServer at a later date

# REST GET

- Browser view for files
  - Base URL: e.g. <http://localhost:8080/>
    - / = Get a list of files
    - Click one to view conversation
    - /load/ = hand enter a file
- WebServices
  - BaseURL: e.g. <http://localhost:8080/ws/>
    - <baseUrl>all
      - Returns a collection of available files as a JSON String
        - {filename, platformcode,description}
      - Fetch file on filename + platformcode
      - Fetch file on platformcode + description

# REST PUT—Submit File

- BaseURL e.g. <http://localhost:8080/ws/>
  - HTTP POST method
  - Web service format same as Browser interface but with authentication information
  - URL format:
    - /put
      - ?fileName=<filename>
      - &platform=<cm or ...>
      - &description=<some description>
      - &username=<login>
      - &password64=<pwd Base64> (optional &password=<pwd>)
      - Cargo (file content) must be part of a Stream



# Rest Remove File

- BaseURL e.g. <http://localhost:8080/ws/>
  - HTTP POST method
  - Web service format same as Browser interface but with authentication information
  - URL format:
    - /remove
      - ?fileName=<filename>
      - &platform=<cm or ...>
      - &username=<login>
      - &password64=<pwd Base64> (optional &password=<pwd>)

# Configuring Jetty

- Default configuration calls for port 8080
  - View at <http://localhost:8080/>
- To reconfigure to port 80:
  - Add `-Djetty.port=80` to the batch file or shell script, e.g.
    - `java -Xms256M -Xmx1024m -Djetty.port=80 -jar start.jar`
  - View at <http://localhost/>

# Configuration—Default Admin

- Servlet Directory Structure

- webapps (dir)
  - ROOT (dir)
    - WEB-INF (dir)
      - classes (dir)
      - lib (dir)
      - templates (dir)
      - core-portal-props.xml
      - logger.properties
      - web.xml

- User-configurable values in core-portal-props.xml

- Required for initial operations

```
<parameter name="DefaultAdminName" value = "testadmin" />
<parameter name="DefaultAdminPwd" value = "testing!" />
<parameter name="DefaultAdminFullName" value = "Tester 1" />
```

# Configuration—Database

- core-portal-props.xml
  - User-configurable values

```
<parameter name="MyDatabaseUser" value = "nexistw" />  
<parameter name="MyDatabasePwd" value = "" />  
<!-- file index -->  
<parameter name="MyDatabase" value="ibisfiles" />  
<!-- users -->  
<parameter name="UserDatabase" value="ibisusers" />
```

# Configuration—Paths

- core-portal-props.xml
  - User-configurable values

```
<parameter name="MyLogPath" value="logger.properties" />
```

```
<parameter name="MyDataPath" value="webapps/ROOT/data/" />
```

```
<parameter name="MyTextIndexPath" value="webapps/ROOT/WEB-INF/index/text/" />
```

# Configuration—Platforms

- core-portal-props.xml
  - User-configurable values
  - These values must appear in REST URLs

```
<list name = "DocumentTypes">  
  <parameter name = "Compendium" value = "cm" />  
  <parameter name = "DebateGraph" value = "dg" />  
  <parameter name = "Deliberatorium" value = "dm" />  
  <parameter name = "Cohere" value = "co" />  
</list>
```

# Web Interface—Importing

## IBIS Server

[Logout](#)

[Load a file...](#)

### File Entry Form

File Name	<input type="text" value="ClimateChangeTop.xml"/>
Platform	<input type="text" value="cm"/>
cm/dg/dm/co	
Description	<input type="text" value="Climate Change"/>
File Text	<pre>&lt; / b a c k g r o u n d &gt;                                 &lt; c o d e r e f s &gt;                                 &lt; / c o d e r e f s &gt;                                 &lt; s h o r t c u t r e f s &gt;                                 &lt; / s h o r t c u t r e f s &gt;                                 &lt; / n o d e &gt;                                 &lt; n o d e   i d = " 1 9 2 1 6 8 1 1 0 7 1 2 4 6 3 7 3 1 6 4 1 9 8 "   t y p e = " 4 "   e x t e n d e d t y p e = " "   o r</pre>
<input type="button" value="Submit"/>	

Note: authentication required to load a file

# Web Interface—File Listing

## IBIS Server

[Logout](#)

[Load a file...](#)

---

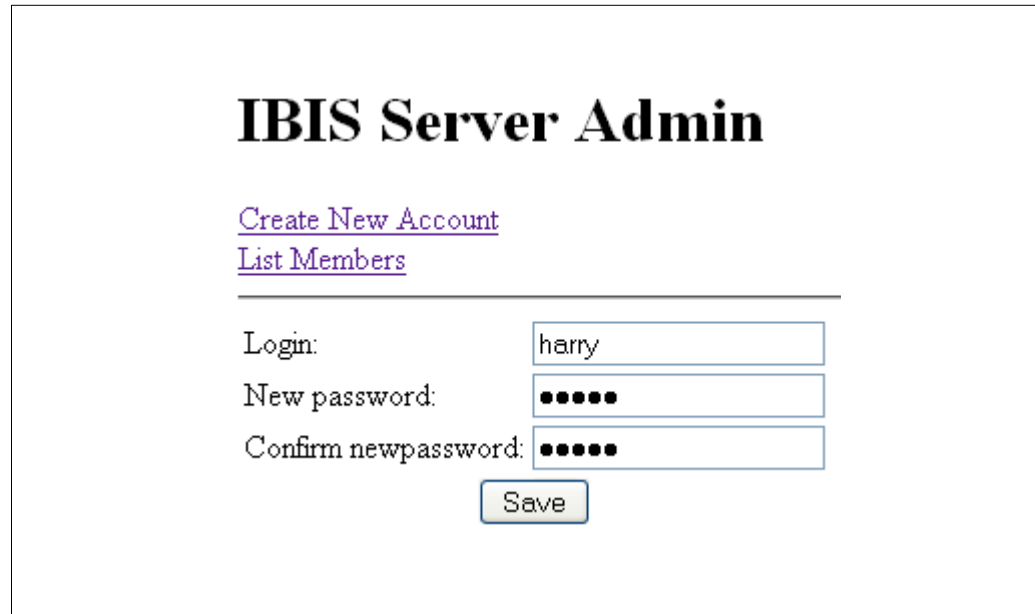
### Files

<a href="#">ggw-3.xml</a>	cm	jackpark	Grains Gone Wild
<a href="#">15GrandChallenges.xml</a>	cm	jackpark	Millennium Projects 15 Global Challenges
<a href="#">OilCrisis-1.xml</a>	cm	jackpark	Description of Oil Crisis
<a href="#">Thermohaline-1.xml</a>	cm	jackpark	Thermohaline Circulation



# Web Interface—New Member

- Used for creating new IBISServer Users



The screenshot shows a web interface titled "IBIS Server Admin". Below the title are two links: "Create New Account" and "List Members". A horizontal line separates the links from the form fields. The form includes three input fields: "Login:" with the value "harry", "New password:" with five dots, and "Confirm newpassword:" with five dots. A "Save" button is located below the "Confirm newpassword:" field.

**IBIS Server Admin**

[Create New Account](#)  
[List Members](#)

---

Login:

New password:

Confirm newpassword:

Admin Interface requires admin authentication

# Web Interface—Authentication

## IBIS Server

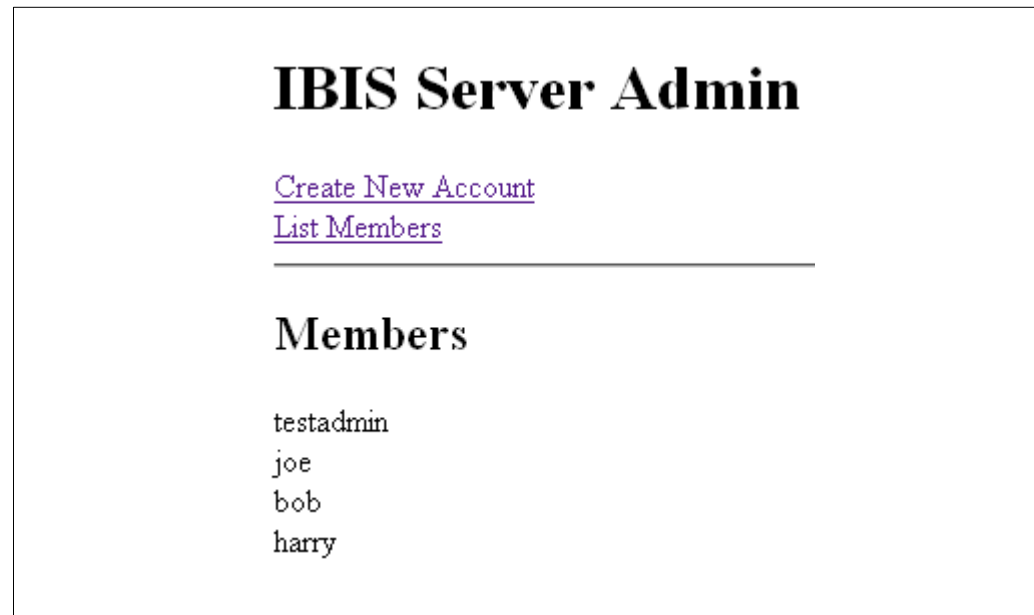
[Login](#)

---

Username

Password

# Web Interface—List Members



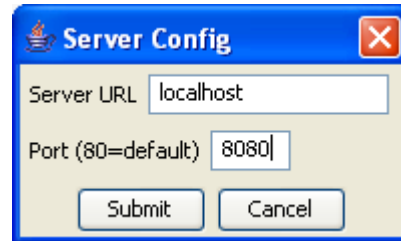
Admin Interface requires admin authentication

# Desktop Client Platform

- Standalone Java platform
  - Uses Apache HTTPClient
  - Mimics Web services calls
  - Permits uploading (posting) and downloading IBIS conversations from IBISServer

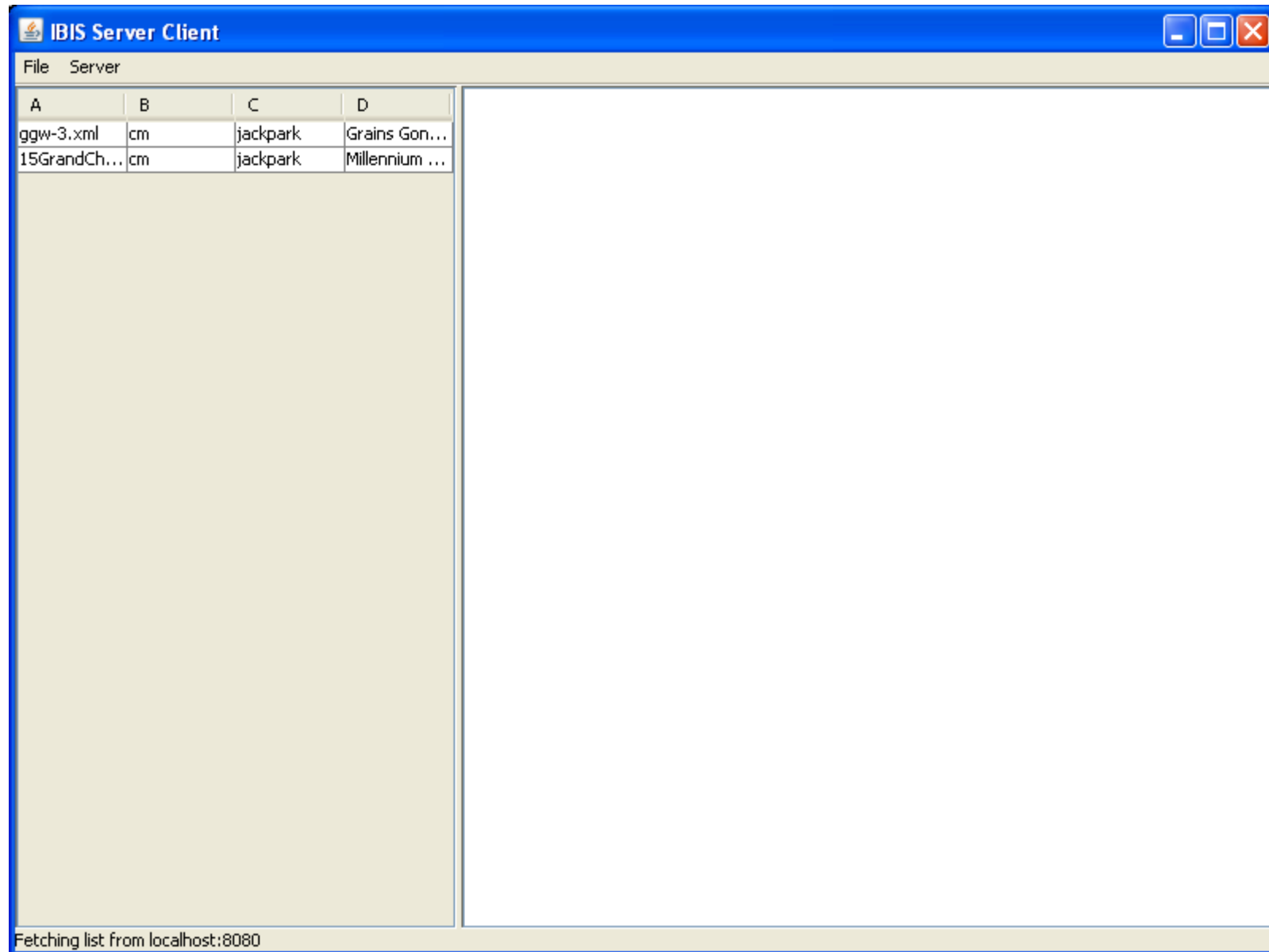
# Starting the Desktop Client

- Configure the Server settings
  - `localhost` used in image below
  - e.g. `http://server.org`



# List Available Files

- Use the menu `Server:List`



# Fetch a File

- Double-click selection

The screenshot shows the IBIS Server Client window. On the left, there is a table with four columns: A, B, C, and D. The first two rows are visible: 'ggw-3.xml' with 'cm' in column B and 'jackpark' in column C, and '15GrandCh...' with 'cm' in column B and 'jackpark' in column C. The right pane displays an XML document. The status bar at the bottom indicates 'Fetching list from localhost:8080'.

A	B	C	D
ggw-3.xml	cm	jackpark	Grains Gon...
15GrandCh...	cm	jackpark	Millennium ...

```
5 2 5 3 1 " lastModified = " 124884035
2 5 3 1 " author = " Jack Park " type =
" 39 " originalid = " " from = " 19216
8 1 1 0 2 1248840350328 " to = " 192168
1 1 0 2 1248840077734 " label = " " ar
row = " 1 " >

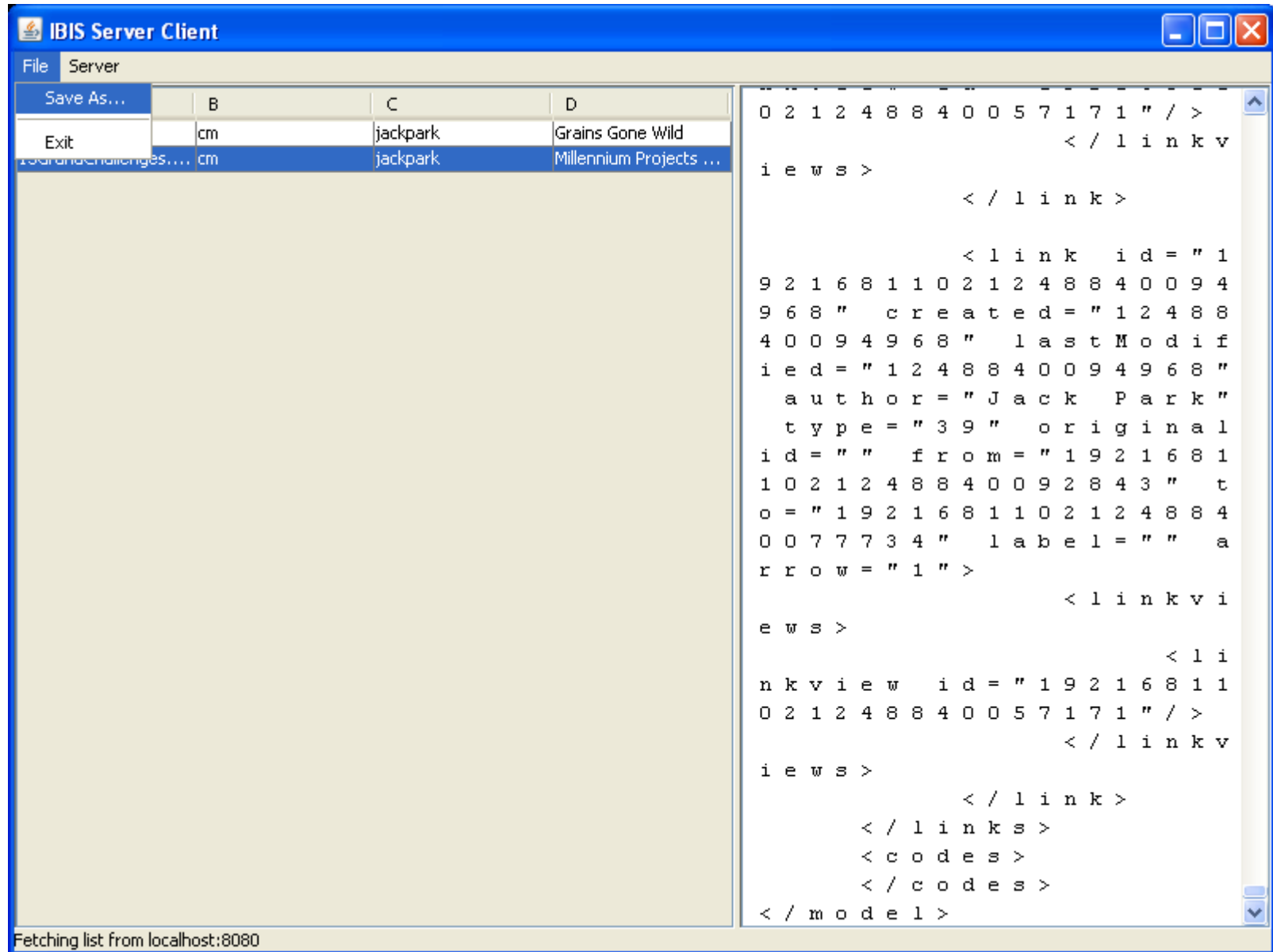
    < linkviews >
        < linkview id = "
19216811021248840057171" / >
    < / linkviews >
< / link >

    < link id = " 192168110212
48840094968 " created = " 12488400
94968 " lastModified = " 124884009
4968 " author = " Jack Park " type =
" 39 " originalid = " " from = " 19216
8 1 1 0 2 1248840092843 " to = " 192168
1 1 0 2 1248840077734 " label = " " ar
row = " 1 " >

    < linkviews >
        < linkview id = "
19216811021248840057171" / >
    < / linkviews >
< / link >
< / links >
< codes >
< / codes >
< / model >
```

Fetching list from localhost:8080

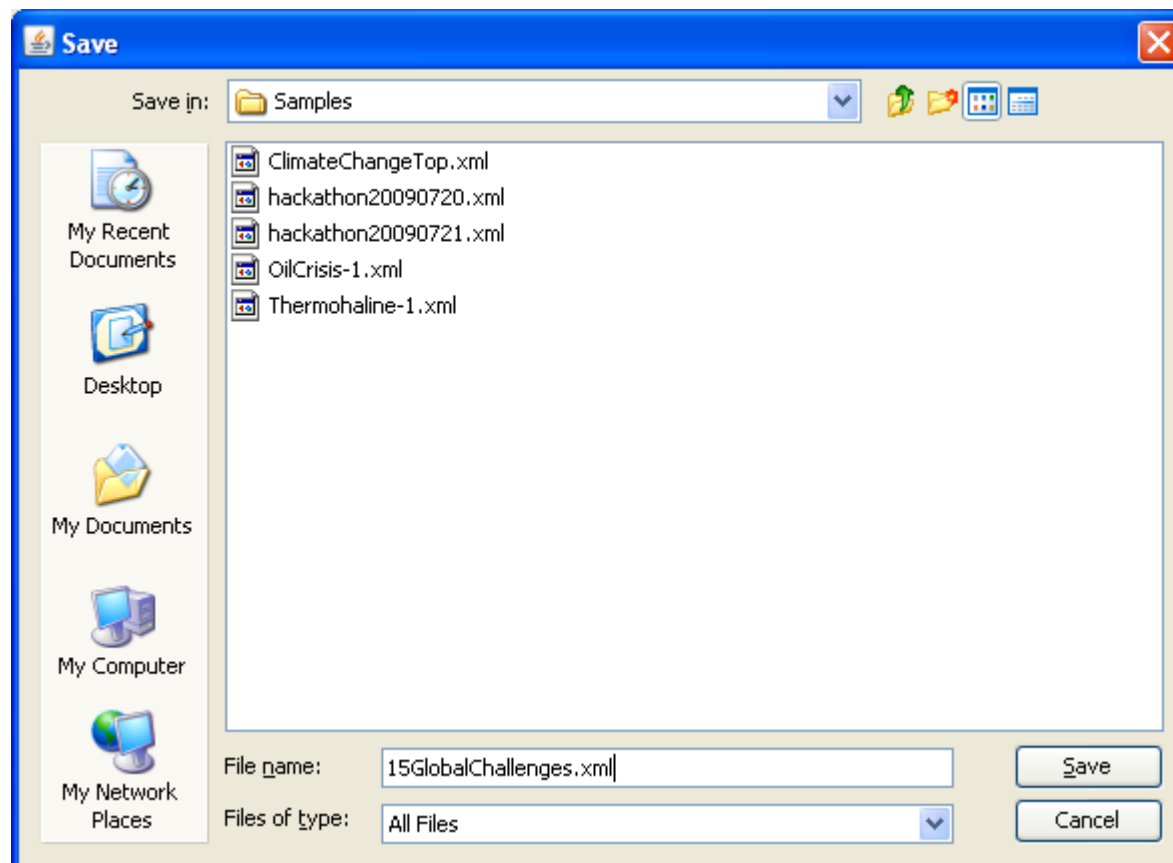
# Saving Downloaded File—1





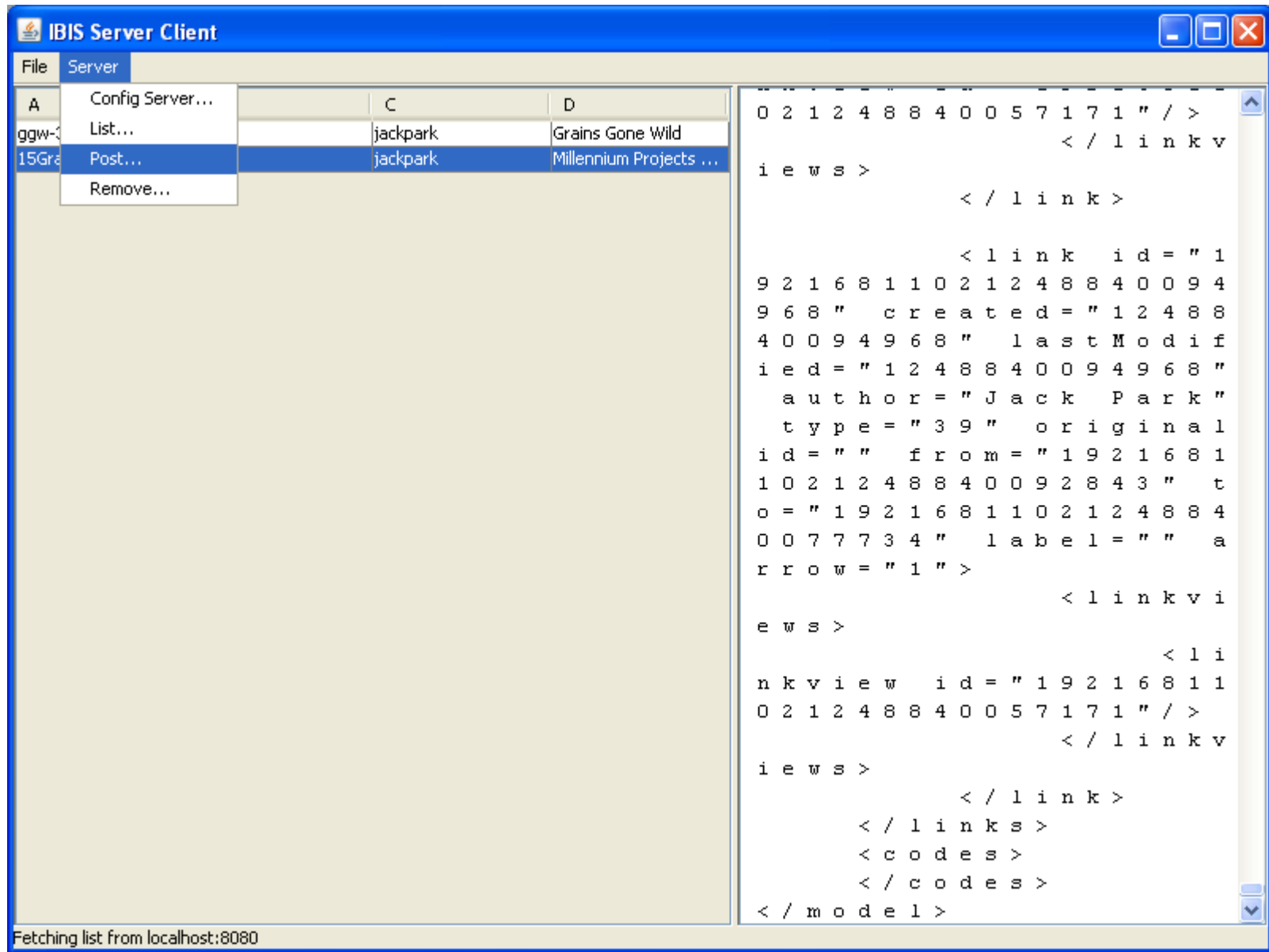
# Saving Downloaded File—2

- Name the file and click Save



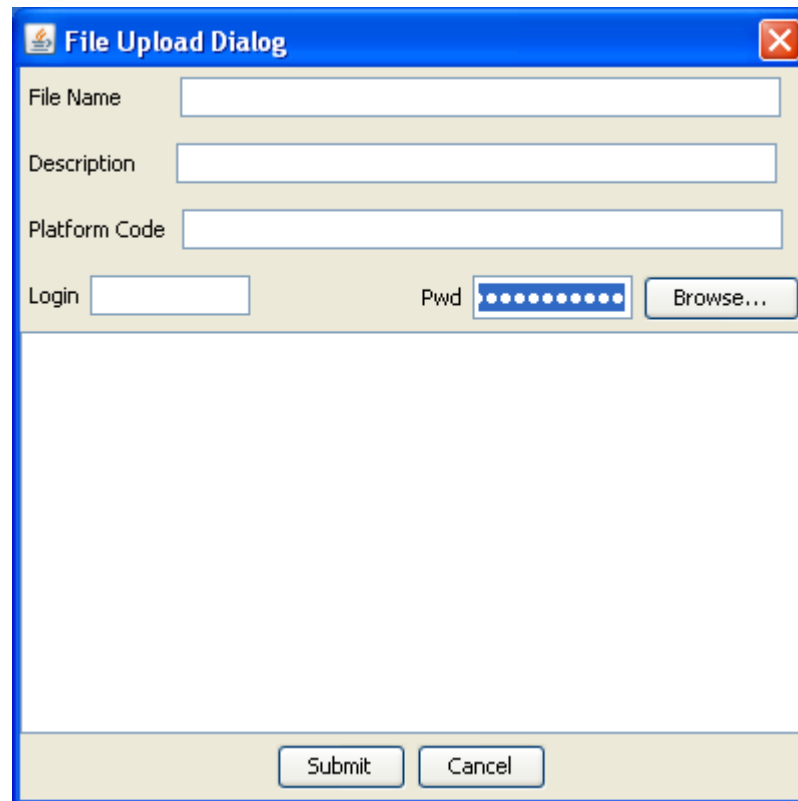
# Posting—1

- Use menu `Server:Post`



# Posting—2

- Click Browse button

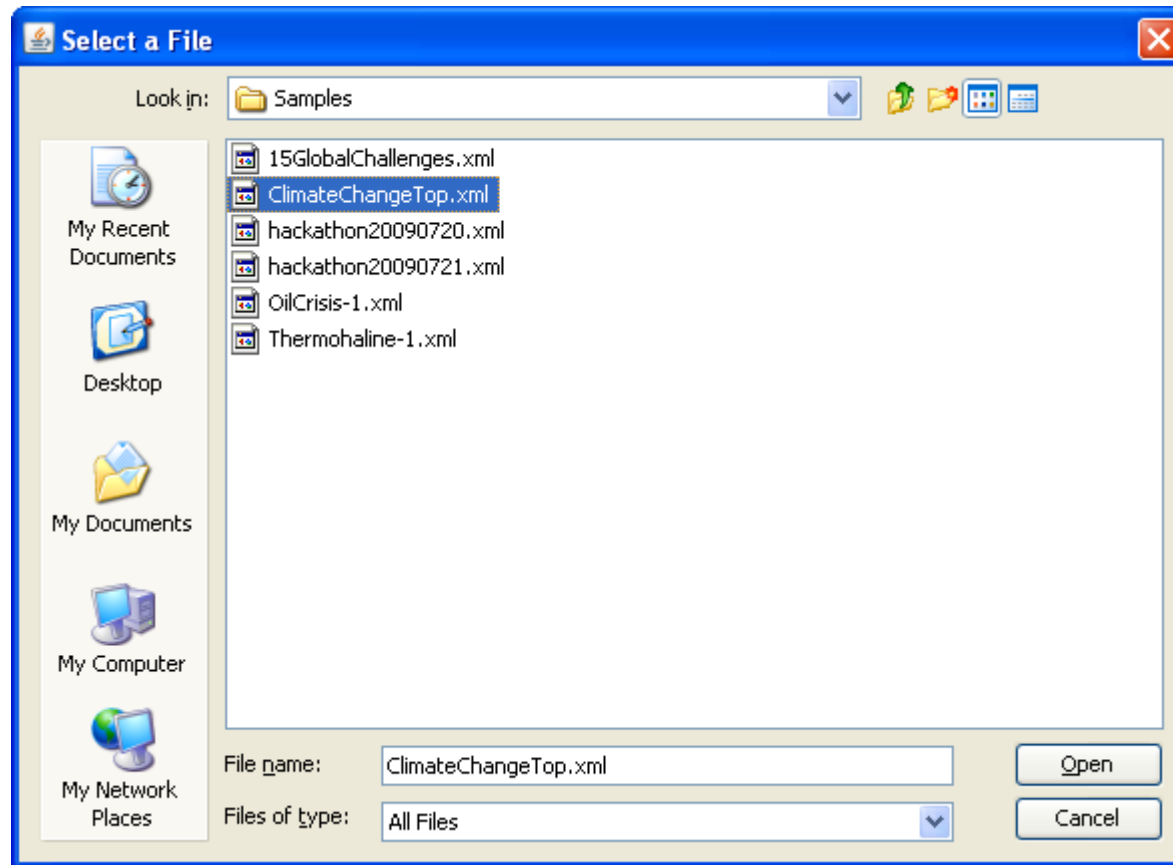


A screenshot of a 'File Upload Dialog' window. The window has a blue title bar with the text 'File Upload Dialog' and a close button. The main area is light beige and contains several input fields: 'File Name', 'Description', and 'Platform Code', each with a text box. Below these are 'Login' and 'Pwd' fields, both with text boxes. The 'Pwd' field is masked with dots. To the right of the 'Pwd' field is a 'Browse...' button. At the bottom of the window are 'Submit' and 'Cancel' buttons.

File Upload Dialog	
File Name	<input type="text"/>
Description	<input type="text"/>
Platform Code	<input type="text"/>
Login	<input type="text"/>
Pwd	<input type="password"/>
<input type="button" value="Browse..."/>	
<input type="button" value="Submit"/> <input type="button" value="Cancel"/>	

# Posting—3

- Select a file to post



# Posting—4

- **Fill in** Description, Platform Code, Login **and** Password, **then click** Submit

**File Upload Dialog**

File Name: Thermohaline-1.xml

Description: Thermohaline Circulation

Platform Code: cm

Login: jackpark

Pwd: ...

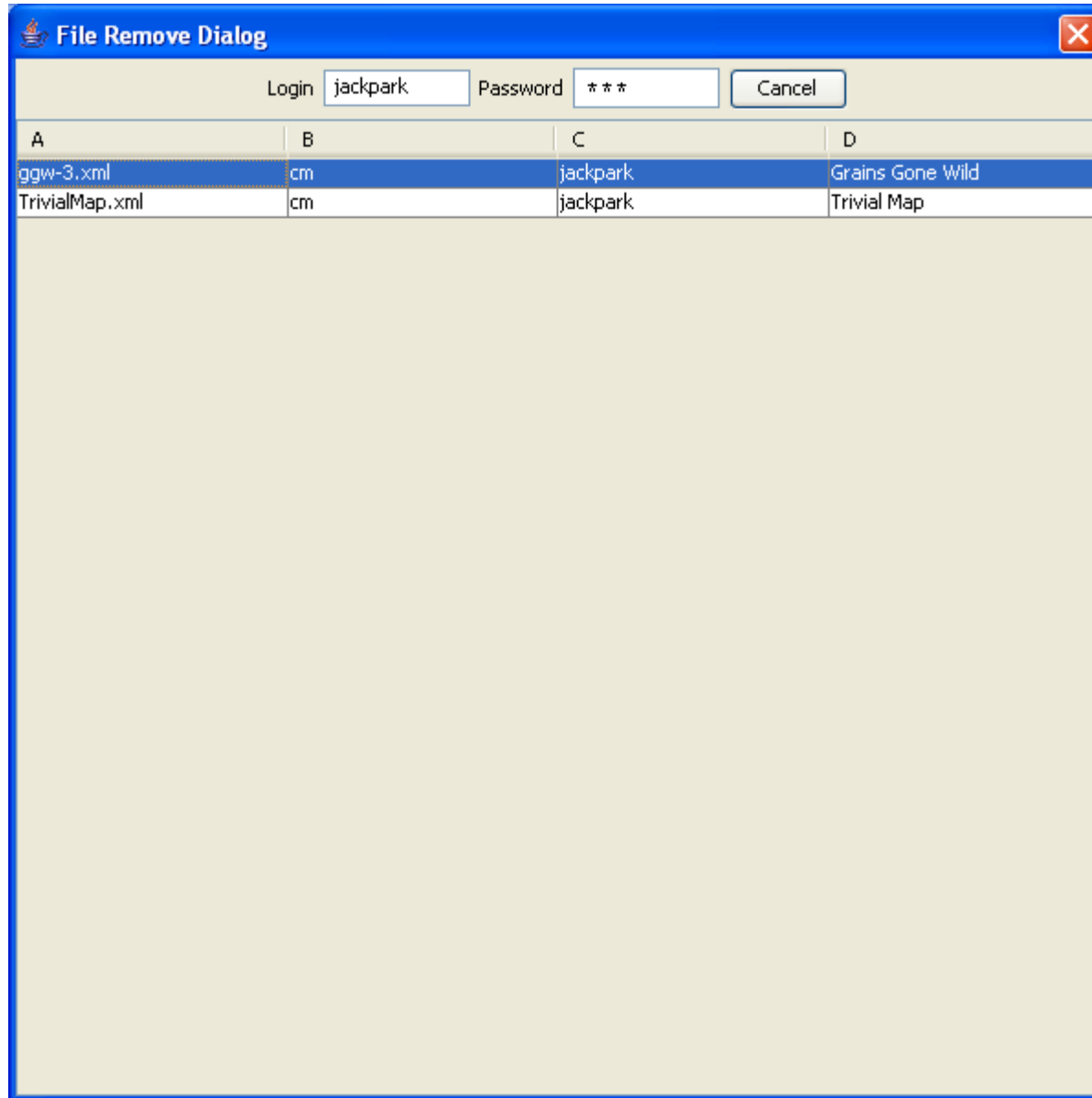
Browse...

0?0x0m0l0 0v0e0r0s0i0o0n0=0"010.000"0 0e0n0c0o  
 0d0i0n0g0=0"0U0T0F0-01060"0?0>0  
 0<0!0D00C0T0Y0P0E0 0m0o0d0e0l0 0S0Y0S0T0E0M0  
 0"0C0o0m0p0e0n0d0i0w0r0.0d0t0d0"0>0  
 0<0m0o0d0e0l0 0r0o0o0t0v0i0e0w0=0"010902010608  
 01010007010204050207050907010504060"0>0  
 0 0<0v0i0e0w0s0>0  
 0 0 0<0v0i0e0w0 0v0i0e0w0r0e0f0=0"  
 0109020106080101000701020405020705090701050406  
 0"0 0n0o0d0e0r0e0f0=0"010902010608010100070102  
 04050305020500010907010"0 0X0P0o0s0i0t0i0o0n0=

Submit Cancel

Note that UTF-16 text does not always appear as text.

# TestPlatform—Remove File



A screenshot of a 'File Remove Dialog' window. The window has a blue title bar with the text 'File Remove Dialog' and a red close button. Below the title bar is a login section with two text boxes: 'Login' containing 'jackpark' and 'Password' containing '\*\*\*'. To the right of these boxes is a 'Cancel' button. Below the login section is a table with four columns labeled A, B, C, and D. The table contains two rows of data. The first row has 'ggw-3.xml' in column A, 'cm' in column B, 'jackpark' in column C, and 'Grains Gone Wild' in column D. The second row has 'TrivialMap.xml' in column A, 'cm' in column B, 'jackpark' in column C, and 'Trivial Map' in column D. The table is highlighted with a blue border. Below the table is a large, empty, light beige area.

A	B	C	D
ggw-3.xml	cm	jackpark	Grains Gone Wild
TrivialMap.xml	cm	jackpark	Trivial Map

User must fill in  
Login and  
Password  
before clicking a  
file to remove

# Tested in Version 0.1

- Web
  - Admin can create users
  - Users can login
  - Users can upload (copy and paste into a browser)
  - Users can view a selected file as XML
- Desktop
  - List available files
  - Upload new file
  - Download selected file
  - Remove selected file from the server

# Current Implementation Issues

- Not tested with International (e.g. Chinese) files
- Some unresolved issues related to UTF-8 and UTF-16
  - Desktop platform is hardwired for UTF-8 in Version 0.1



# Version 0.2 Plans

- Resolve UTF-8/16 Issues and test on international character files
- Cleaner user interface
- Addition of the IBIS Conversation viewer adapted from the Hypercube project