



Using the Javascript Console for development and administration

Florian Maul (fme AG)

About me



- Florian Maul
 - Senior Consultant at fme AG (Platinum Partner)
 - Many years of experience with Alfresco
 - Winner of the Dashlet Challenge 2011 Gallery Plus Dashlet
 - Opensource Contributor to Share Extras and other Alfresco Extensions
 - Android CMIS Browser (2010)



@fmaul

<http://alfresco.fme.de/>

Agenda

- Overview and History
- while(t < 50) {
 - Javascript Console Features
 - Demos
- }

Alfresco APIs

- What kinds of APIs are there?
 - Java Foundation API
 - Javascript API & Javascript Services
 - CMIS
- Where can I use Javascript in Alfresco:
 - Webscripts developed in Javascript
 - Javascript run by Actions (Run Script Action)
 - Javascript attached to Tasks & Events in Workflows

But how to develop Javascript code?

- Webscripts
 - Develop in Eclipse
 - ANT build script to copy changes & reload the webscripts
 - Run the webscript manually in the browser
- Data Dictionary Scripts
 - upload a script to Data Dictionary
 - run Script Action
 - watch errors in the Alfresco Log file (ScriptLogger=DEBUG)
 - fix bugs and iterate

→ It takes too long to test Javascript code and fix bugs.

Meet the Javascript Console



My Dashboard Sites People Repository More... Devtools

Administrator Help

Enter search terms...

Admin Console

Tools

Javascript Console

Application

Category Manager

Node Browser

Tag Manager

File Management

Trashcan

Content Publishing

Channel Manager

Repository

Activities Feed

Fileservers

Google Docs

License Descriptor

Replication Jobs

Repository Descriptor
(Originally Installed)

Repository Descriptor
(Current)

Repository Tools

run with **var space** = /Company Home

Select ...

Docs

Save Script

Load Script

Javascript input

Freemarker input

Script execution parameters

```
1 var nodes = search.luceneSearch("@name:alfresco");
2
3 for each(var node in nodes) {
4     logger.log(node.name + " (" + node.typeShort + "): " + node.nodeRef);
5 }
6
```

Execute

or Ctrl+Enter

Console output

Freemarker HTML output

Freemarker text output

```
alfresco (cm:folder): workspace://SpacesStore/bf62e1ef-c768-4872-90c0-7c2f4c592db5
GROUP_ALFRESCO_ADMINISTRATORS (cm:authorityContainer): workspace://SpacesStore/GROUP_ALFRESCO_ADMINISTRATORS
alfresco-teaser.png (cm:content): workspace://SpacesStore/7927181f-a9f0-4893-bdb7-59968b69c728
fme-alfresco-modules (cm:folder): workspace://SpacesStore/abc9b08e-c600-44c8-8da4-0c7d773901b6
alfresco docs.js.sample (cm:content): workspace://SpacesStore/cdb970e0-39fe-4a1c-b918-2c0b9db1dc47
Datenblatt Alfresco by fme.pdf (cm:content): workspace://SpacesStore/c9a04983-f452-4b4d-9f08-06d965bc97fa
fme-alfresco-modules discussion (fm:forum): workspace://SpacesStore/c16e5499-c5b3-4f24-bd1b-e4f829fb7b97
emailbody_textplain_alfresco.ftl (cm:content): workspace://SpacesStore/9a8b55e7-8f32-4c25-94cd-c60b8839647e
emailbody_textplain_alfresco_it.ftl (cm:content): workspace://SpacesStore/f6232a8b-201d-4d00-af99-1f2b9a9f7a55
emailbody_texthtml_alfresco.ftl (cm:content): workspace://SpacesStore/e388f3d2-5842-4104-935f-b1a806425a6a
emailbody_texthtml_alfresco_it.ftl (cm:content): workspace://SpacesStore/34cb014b-d3ce-41ee-b3b4-65fb3d1c2ec0
Alfresco-Paket 2012.pdf (cm:content): workspace://SpacesStore/afae0f43-8b92-4bf6-853f-ea116e8de3bf
emailbody_textplain_alfresco_de.ftl (cm:content): workspace://SpacesStore/8f0009ea-4f17-4476-8591-238d1687cd18
```


Meet the Javascript Console



- **Mike Farman (Director Product Management Alfresco):**
„I really like the stuff you’ve been doing, my favourite is the JavaScript console, it’s the first add-on I always install!“



Will Abson
@wabson

Following



Looking at Share Extras top downloads so far in 2012 - [@fmaul](#)'s wonderful JavaScript Console is leading the pack [#Alfresco](#)

← Reply ↻ Retweeted ★ Favorite

Javascript Console Features













- Component for Share Admin Console to develop, execute and test Javascript code
- Features
 - Full access to the Alfresco JavaScript API
 - Specify a context (space, document, user)
 - Simple code completion
 - Load & save scripts from the repository
 - Import Javascript code from the Alfresco classpath
 - Develop freemarker templates (since v0.5)

Javascript Console Milestones




- End of 2010
 - Need to develop and execute Javascript code for a project sparked development of the first prototype
 - First internal versions with a single standalone page in Share
- April 2011:
 - Contact with Will Abson
 - refactoring and integration into the admin console of Alfresco 3.4
- May 2011:
 - First checkin into the Share-Extras project and first public release
- Nov 2011:
 - Version 0.4 with basic code completion
- June 2012:
 - Version 0.5 with freemarker editor

Download & Installation





- Download from <http://code.google.com/p/share-extras/>

		ical-feed-dashlet-0.2.jar	iCal Feed Dashlet add-on for 3.3/3.4	Featured
		ical-feed-dashlet-2.0.jar	iCal Feed Dashlet add-on for 4.0	Featured
		javascript-console-0.5.zip	Javascript Console add-on JAR for 3.4/4.0	Featured
		linkedin-dashlet-2.1.1.jar	LinkedIn dashlet add-on JAR for Alfresco 4.0	Featured
		media-preview-0.2.jar	Media Preview add-on for 3.3/3.4	Featured
		media-viewers-2.0.jar	Media Viewers add-on for 4.0	

- 3.4.x or 4.0.x

Name
 3.4.x
 4.0.x
 README

AMP or JAR versions

Name	Typ
 javascript-console-repo-0.5.amp	AMP-Datei
 javascript-console-repo-0.5.jar	Executable Jar File
 javascript-console-share-0.5.amp	AMP-Datei
 javascript-console-share-0.5.jar	Executable Jar File

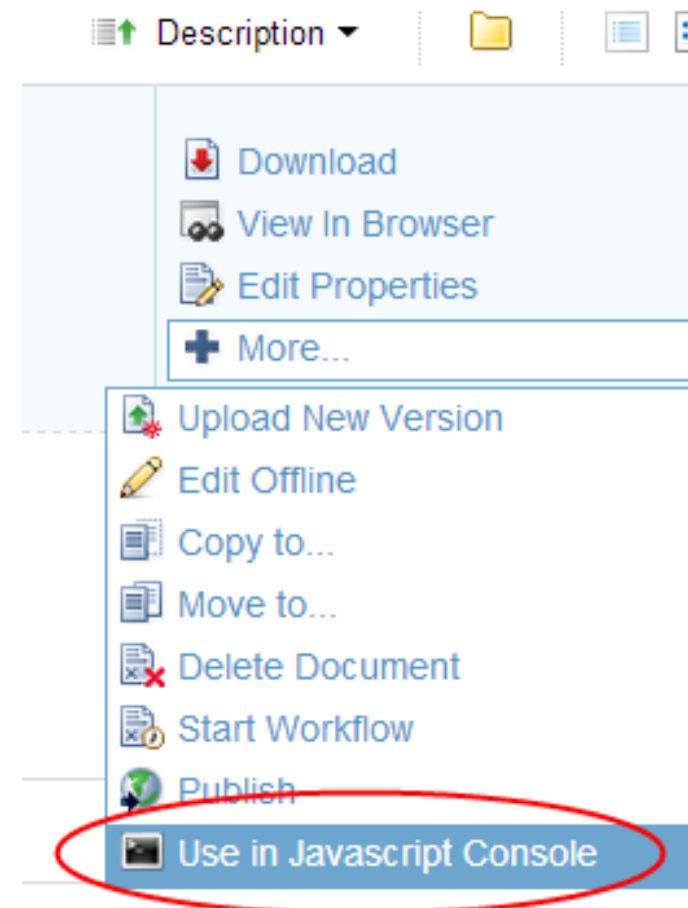
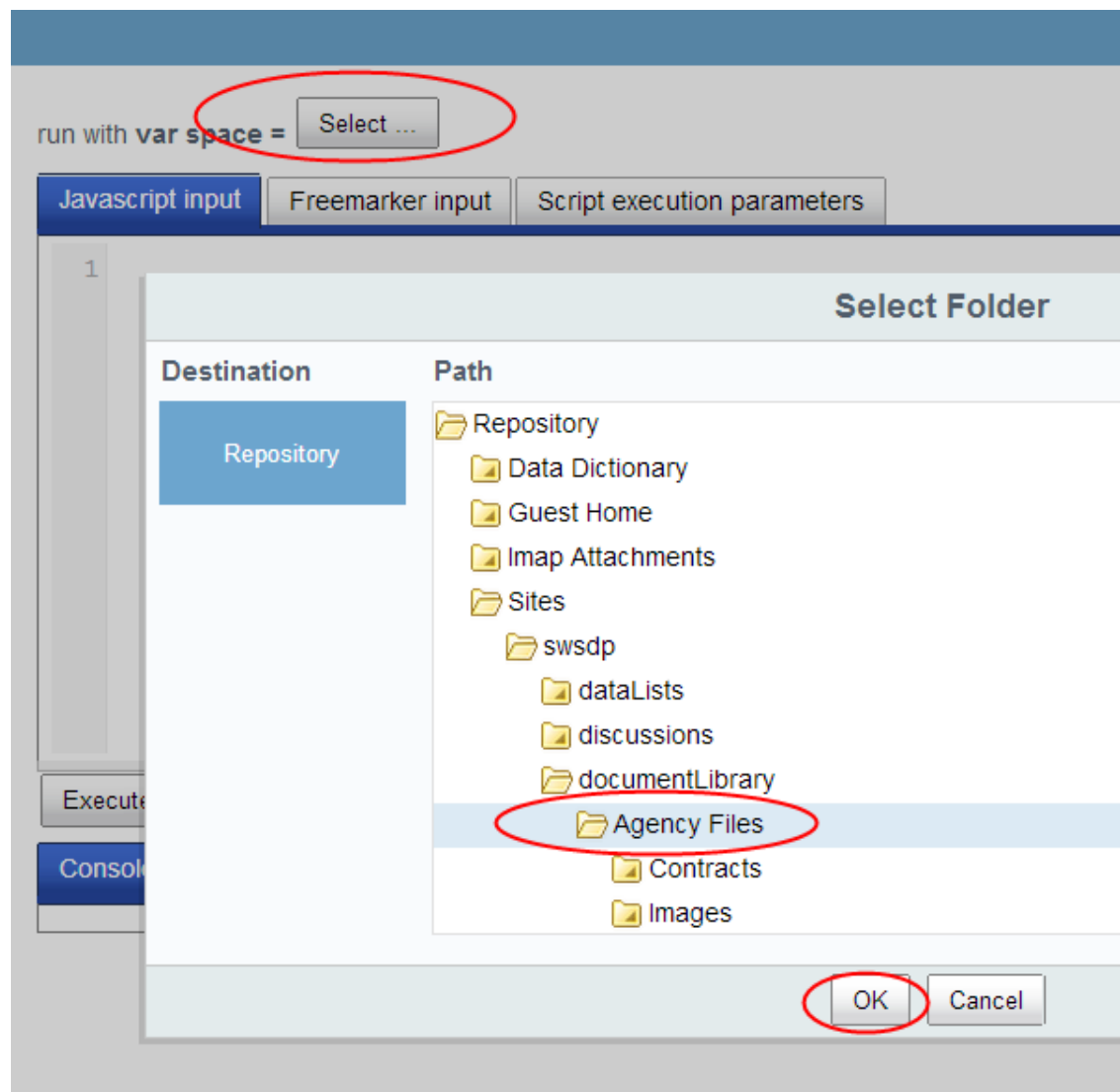
- The repo.jar must be copied to tomcat/webapps/alfresco/WEB-INF/lib and the share.jar to tomcat/webapps/share/WEB-INF/lib

Basic Demo

- Overview
- „Poor mans“ Nodebrowser

Selecting the file or folder to work with

- Select the space variable from a directory chooser:
- Open any document in the Javascript Console from the document library:



Custom Javascript functions

- `print()`
 - Prints a String, Number or an Array to the console output window
 - Formats output for `ScriptNode`, `NodeRef`, `ScriptUser`, `ScriptGroup`, `ChildAssociationRef` objects
- `logger.log()`
 - Logs normally to the debug log but also performs a `print()`
 - Recommended for any output! (works outside of JS Console)
- `logger.setLevel()`
 - Sets the log4j log level
 - `logger.setLevel(„org.alfresco.repo.jscript.ScriptLogger“, „DEBUG“);`
- space variable can be changed from code
 - e.g. `space = space.parent;`
 - On the next script run, the new value will be used.

Custom Javascript functions

- `recurse()`
 - Special function to iterate over a tree of nodes:

```
recurse(space, function(node) {  
    if (node.name.toLowerCase().indexOf("alfresco") >= 0) {  
        print("adding tag alfresco to " +node.displayPath+"/"+node.name);  
        node.addTag("alfresco");  
    }  
});
```

- If no function is given it returns the nodes as an array:

```
var allNodes = recurse(space);  
print(allNodes);
```

- Source: [javascript-console-repo/config/de/fme/jsconsole/jsconsole-post-roll-script.js](#)

Demo – User Management

- Creating users

```
var p = people.createPerson(username,  
    firstname, lastname,  
    email, password, true);
```


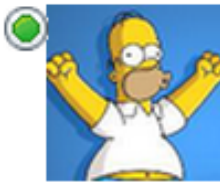



```
p.properties["cm:location"] = location;  
p.properties["cm:jobtitle"] = jobtitle;
```

- and attaching avatars

```
for each (imageNode in space.children) {  
    var user = people.getPerson(name);  
    user.createAssociation(  
        imageNode, "cm:avatar");  
}
```

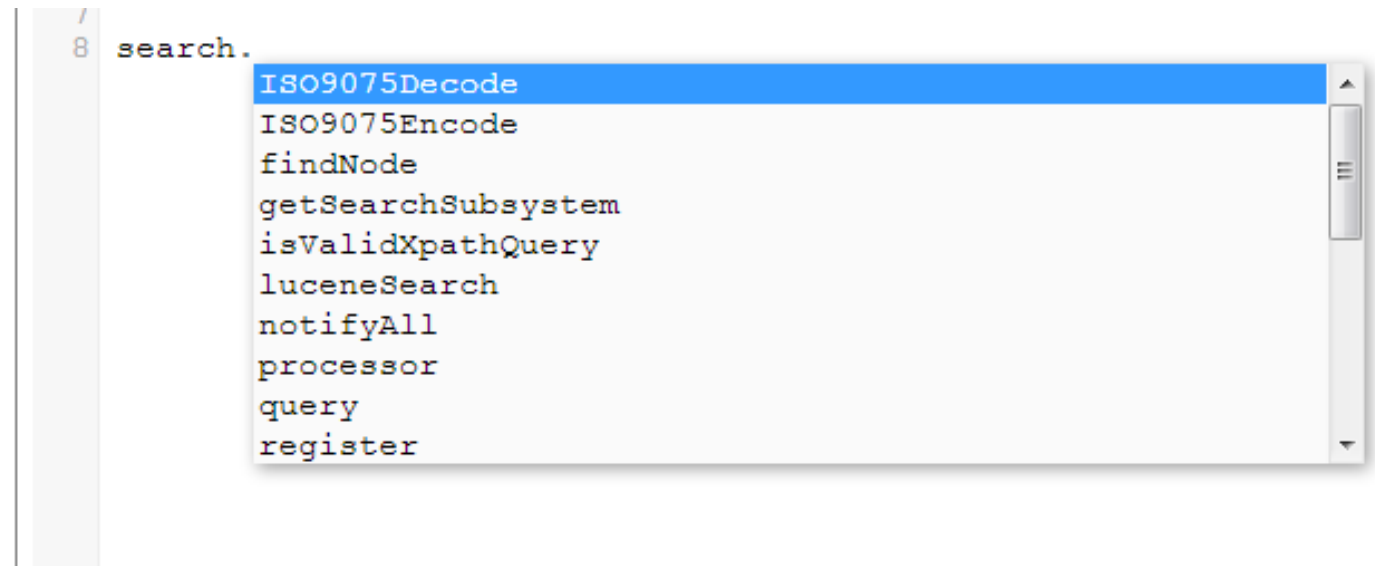
User Search

Search for "simp*" found 5 results.

	Name	User Name
	Bart Simpson	bart
	Homer Simpson	homer
	Lisa Simpson	lisa
	Maggie Simpson	maggie
	Marge Simpson	marge

Code completion

- Use with Ctrl+Space
- Javascript operations
 - `for`, `if`, `while`, ...
- All javascript root objects
 - `search`, `siteService`, ...
- Custom javascript extensions are included as well
- All methods 1 level below the root objects
 - `search.findNode()`, `siteService.getSite()`,...
- Some variables are completed as ScriptNode objects:
 - `space`
 - `document`
 - `companyhome`
 - `*node` or `*Node`



Demo - Tagging

- Automatically tag content
- Spreadsheet metadata import

How does it work?

Alfresco Repository / Webscripts

Execute Webscript

- print(), recurse(), logger
- eval() / Java-Webscript
- <imports>
- Error handling

JSON POST

Browser / Javascript / YUI / Codemirror

Input

- Javascript code
- Freemarker template
- space / document

Output

- Print / log output
- Freemarker output
- Error messages

Demo - permission reporting

- All parent permission

```
while (document) {  
    print(document);  
    print(document.fullPermissions);  
    print("");  
    document = document.parent;  
}
```

- All direct permissions

```
recurse(space, function(node) {  
    for each(permission in node.fullPermissions) {  
        if (//;DIRECT$/.test(permission)) {  
            logger.log(node.displayPath + "/" + node.name + ";" + permission);  
        }  
    }  
});
```

Demo - Creating Webscripts

- The new version 0.5 makes it easy to create webscripts:
 - Write a controller javascript using logger.log output
 - Create a freemarker template
 - See HTML or Text output
 - Test different URL arguments and different users

→ DEMO

The screenshot shows the Alfresco web script editor interface. At the top, there are three tabs: 'Javascript input', 'Freemarker input' (which is selected), and 'Script execution parameters'. Below the tabs is a text area containing the following code:

```
1 [
2   <#list folder as f>
3     {
4       "name" : "${f.name}",
5       "creator" : "${f.creator}",
6       "creationDate" : "${f.creationDate?date}"
7     }
8   </#list>
9 ]
```

Below the code area is an 'Execute' button with the text 'or Ctrl+Enter' next to it. At the bottom, there are three tabs: 'Console output', 'Freemarker HTML output' (which is selected), and 'Freemarker text output'. Below these tabs is a text area showing the rendered output:

```
[
  {
    "name" : "Email Templates",
    "creator" : "System",
    "creationDate" : "May 18, 2012"
  },
  {
    "name" : "Imap Configs",
    "creator" : "System",
    "creationDate" : "May 18, 2012"
  }
]
```

The bottom right corner of the slide features the 'Alfresco.org Devcon 2012' logo.

Keyboard shortcuts

- Currently supported

- Ctrl + Enter Execute Script
- Ctrl + Space Code Completion
- Ctrl + Z Undo
- Ctrl + Y Redo

- Upcomming (0.5.1)

- Ctrl + Shift + F Format javascript code (beautify.js)
- Ctrl + / Comment / uncomment with //

Access spring beans / unlock documents

- LockService is not accessible through the Javascript API
- Example to unlock a document:

```
var ctx = Packages.org.springframework.  
web.context.ContextLoader.getCurrentWebApplicationContext();
```

```
var lockService = ctx.getBean("lockService");
```

```
lockService.unlock(document.nodeRef);
```

**With great power
comes great
responsibility!**

- may look like a hack but is perfectly valid Alfresco JS
- It only works in Javascript stored in the Classpath (not in the Data Dictionary)
- Also very useful to test your own Beans written in Java

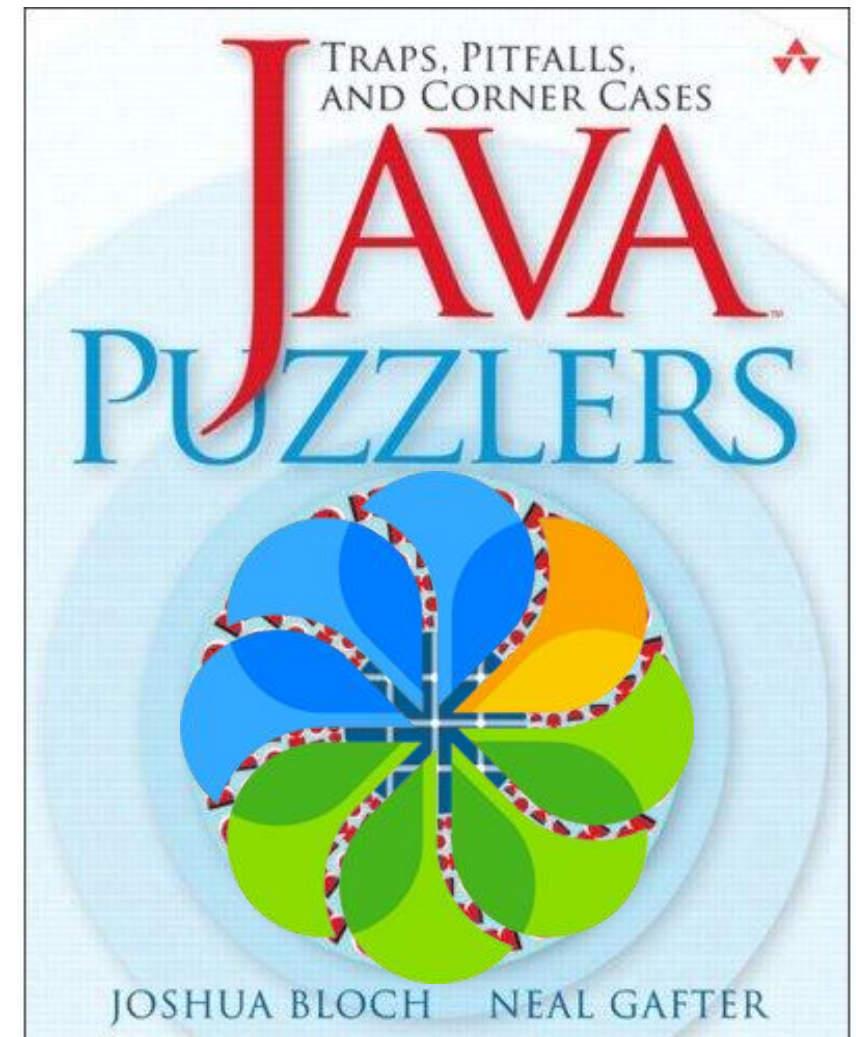
Javascript Puzzlers – Alfresco Edition

What do these two do?

- `node.properties["cm:name"] = "textfile.txt";`
- `node.name = "textfile.txt";`

Comparing ScriptNode nodeRefs:

- `node1.nodeRef.equals(node2.nodeRef)`
- `node1.nodeRef == node2.nodeRef`
- `" " + node1.nodeRef == " " + node2.nodeRef`



Batch processing

Examples

- Set all documents to inherit permissions
- Disable automatic versioning for all documents

Small batches can run in the Javascript Console

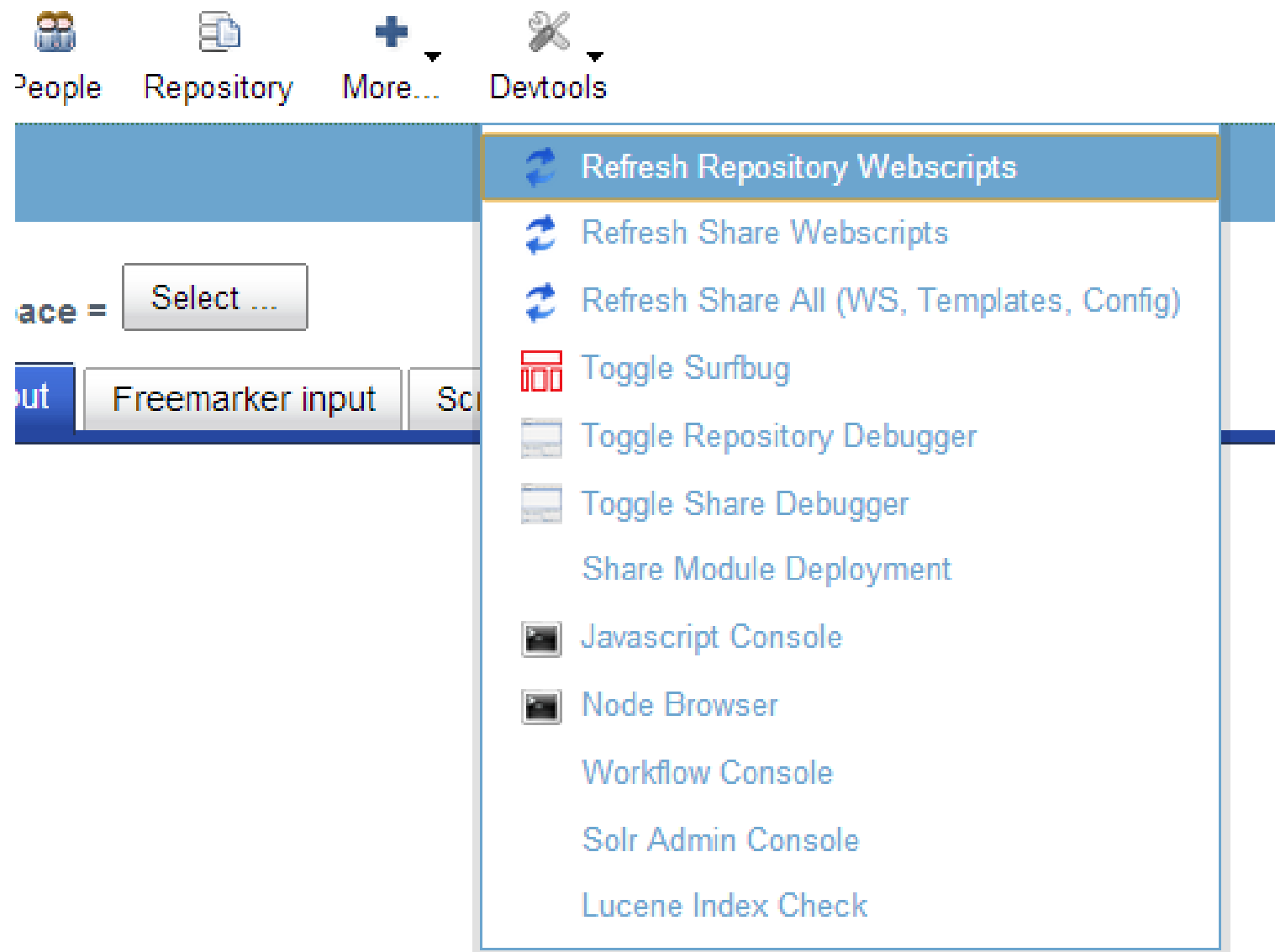
- Single transaction (up to about 10000 nodes)
- I install the Javascript Console on every production system

Large batches currently only possible using Java

- Batch processing (updates for >100000 nodes)
- How to find the nodes? Search or recursion?
- Split in smaller batches with own transaction
- How to make use of multithreading?

Developer Tools Menu

- If you like the Javascript Console, you might also like...



<http://code.google.com/p/fme-alfresco-extensions/wiki/DeveloperToolsMenu>

Questions?