



Documentation for OpenKM 6.3 - CE

Table of Contents

Table of Contents	2
OpenKM 6.3 - CE	34
Hardware and software requirements	36
Requirements	36
Checking the Java version	36
For Windows users	36
Hardware configuration	37
Browser support	38
Operating systems	38
Third-party software requirements	40
Installation	42
Installing on Ubuntu and Debian	44
Preliminaries	44
Check disk size	44
Create user named openkm	44
Check if partners channels are enabled	44
Check server configuration	44
Increase ulimit	44
Checking Java version	45
Install Java	45
Set OpenJDK 1.8 as your default Java version	45
Installing database	46
Change MySQL root password	46
Method 1	46
Method 2	46
Method 3	46
Method 4	46
Check InnoDB is default MySQL engine	47
Create database	47
Installing OpenKM and Tomcat bundle	47
Configuring Tomcat as a service	48
Check the service	49
Installing third-party software	49
Start application	49
Check OpenKM.cfg parameters	50
Other common additional configurations setup	50
Restart installation from scratch	51
Installing on RedHat and CentOS	52
Preliminaries	52
Check disk size	52
Create a user named openkm	52
Check server configuration	52
Increase ulimit	52
Checking Java version	53
Install Java	53
Set OpenJDK 1.8 as your default Java version	53
Installing database	53
Change MySQL root password	54
Method 1	54
Method 2	54
Method 3	54
Method 4	54
Check InnoDB is default MySQL engine	55
Create database	55
Installing OpenKM and Tomcat bundle	55
Configuring Tomcat as a service	56
Check the service	57
Installing thirdparty software	57
Start application	58
Check OpenKM.cfg parameters	58

Linux Oracle 7.X Troubleshooting	59
Install libreoffice from RPM	59
Install Tesseract	59
Other common additional setup configurations	60
Restart installation from scratch	61
Installing on Windows	62
Preliminaries	62
Checking Java version	62
Install Java	62
Installing database	62
Check InnoDB is default MySQL engine	63
Create database	63
Installing OpenKM and Tomcat bundle	63
Configuring Tomcat as a service	63
Check the service	67
From terminal	67
From Windows service panel	67
From tomcat7w tool	67
Configuring environment variables	68
Installing OpenKM	71
Start application	72
Check OpenKM.cfg parameters	72
Other common additional setup configurations	73
Restart installation from scratch	73
Using the installer	74
Supported databases	75
Other options	76
Using Docker	77
Database	78
Configure database	78
Configuring MySQL and MariaDB	79
Preliminaries	79
Change default engine	79
Change default charset	79
Database creation	80
Create a database and user	80
Configure your OpenKM.cfg	80
MySQL timezone	80
Configure Tomcat datasources	81
MySQL:	81
MariaDB:	81
Configure application login	81
Check for JDBC Driver	82
Run application	82
Additional information	82
Remove and create MySQL service in Windows	82
Migrate from Windows to Linux	83
MySQL tuning	84
Sample one	84
Sample two	85
Change MySQL data directory	87
Ubuntu	87
CentOS and RHEL	88
Windows	89
MySQL 8 troubleshooting	90
Could not create connection to database server	90
Strong Password Encryption for Authentication	90
Old JDBC driver	92
The server time zone value is unrecognized	93

Problem connecting to your database	93
Configuring Oracle	95
Database creation	95
Create tablespace	95
Create a database and user	95
Configure your OpenKM.cfg	95
Configure Tomcat datasources	95
Configure application login	96
Check for JDBC Driver	96
Run application	96
Configuring PostgreSQL	97
Database creation	97
Create a database and user	97
Configure your OpenKM.cfg	97
Configure Tomcat datasources	97
Configure application login	98
Check for JDBC Driver	98
Run application	98
Additional information	98
PostgreSQL 9.6 and upper	98
Configuring SQL Server	99
Database creation	99
Create a database and user	99
Configure your OpenKM.cfg	100
Configure Tomcat data sources	101
Configure application login	101
Check for JDBC Driver	101
Run application	102
Troubleshooting	102
You can not connect to MS SQL Server	102
Connection lost after some time	102
Not able to login into OpenKM	102
Additional information	103
Configuring other databases	104
Backup and restore	105
Detailed steps	106
Backup	106
Restore	106
Backup scripts	107
Additional information	107
Linux backup and restore with BorgBackup	108
Installation	108
Sample script	108
Backup to remote computer with SSH	108
This is an improved SSH version	109
Backup to mapped network	109
Configure crontab	110
Recovery	111
Linux backup and restore with rdiff-backup tool	112
Debian & Ubuntu	112
CentOS & RedHat	112
Preliminaries	112
Sample script	112
Increase the backup period	115
Do remote backup	115
Do remote backup to SMB or CIFS	116
Do PostgreSQL backup	116
Configure crontab	116
Restoring backup	117
Additional information	117

Linux Sample Backup with Duplicity	119
Installation	119
Command line basics	119
Preliminaries	120
Sample script	120
Increase the backup period	123
Perform the remote backup	123
Perform the remote backup to SMB or CIFS	123
Do the PostgreSQL backup	124
Do the remote backup to Amazon s3	124
Configure crontab	125
Troubleshooting deleting old backups	125
Restoring backup	125
Additional information	126
Linux backup and restore with rsync	127
Preliminaries	127
Sample script	127
Do backup with rotation	129
Do remote backup	130
Do incremental backup with backup-dir	130
Do incremental backup with link-dest	131
Do remote backup to SMB or CIFS	131
Do PostgreSQL backup	131
Configure crontab	132
Rsync and FTP	132
Restoring backup	133
Additional information	133
Linux backup with LFTP	134
Preliminaries	134
Sample script	134
Configure crontab	136
Instance replication	137
Different implementations	138
Sample Linux implementation	138
Setup	140
Antivirus configuration	141
Linux	141
Debian / Ubuntu	141
Red Hat / CentOS	141
Windows	142
Clamd Service Setup	143
Freshclam Virus Signature Auto Update	144
Cleaning older activity log data	145
MySQL	145
Linux	145
Configure crontab	145
Windows	146
Database replication with federated tables	146
Configuring Apache HTTP Reverse-Proxy	148
Internal IP vs external IP	148
Debian and Ubuntu	149
Check the configuration	150
Red Hat and CentOS	150
Check the configuration	151
Windows	151
Max OS X	152
Check the configuration	153
Troubleshooting	153

Additional information	153
Configuring Apache HTTP for changing context URL	154
Additional information	154
Configuring Apache HTTPS Reverse-Proxy	155
Internal IP vs external IP	155
Debian and Ubuntu	156
Redirect all HTTP connections to HTTPS	158
Check the configuration	158
Red Hat and CentOS	158
Check the configuration	159
Troubleshooting	159
Additional information	160
Configuring Nginx HTTP Reverse-Proxy	161
Internal IP vs external IP	161
Debian and Ubuntu	162
Check the configuration	163
Red Hat and CentOS	163
Check the configuration	164
Configuring Lucene Analyzer	166
Configure an Analyzer	166
Configuring OCR engine	168
Tesseract	168
Installation	168
Check installation	169
Configuration	169
Additional information	170
Cuneiform	170
Installation	170
Check installation	171
Configuration	171
Abby OCR for linux	171
Installation	172
Check installation	172
Configuration	172
Dictionary	173
Configuration	173
OCR rotate configuration	173
Configuration	173
Additional information:	173
Configuring OpenSSL	175
Create OpenSSL configuration file	175
Create Certification Authority (CA)	176
Create Apache Certification	177
Apache configuration	177
Certification revocation	178
Add CRL to Apache	179
Configuring SWFTTools	180
Ubuntu	180
Red Hat and Centos	180
Improving performance	181
Debian	181
Improving performance	182
Windows	182
Improving performance	182
Troubleshooting	183
Conversion problems with big files	183
Segmentation fault (core dumped)	183
Error with PDF with copy protected	183

Configuring mail	185
Change mail configuration	185
Gmail example	185
Office 365	185
Additional information	186
Configuring Tomcat memory utilization	187
OpenJDK 1.8	187
OpenJDK 1.7	187
Configuring Tomcat port	188
Change tomcat ports	188
Configuring Tomcat session time-out	190
Configuring Tomcat SSL/TLS	191
Self-signed Certificate	191
Create the Certificate Key store	191
Edit the Tomcat Configuration file	192
Check configuration	192
CA Certificate	192
Prepare the Certificate Key store	192
Create the Certificate Signing Request	192
Install the Root Certificate	193
Install the Intermediate Certificate	193
Install the Primary Certificate	193
Edit the Tomcat Configuration file	193
Check configuration	193
CA Certificate without preparing keystore	194
Obtain p12 certificate	194
Install p12 Certificate	194
Change alias to Primary Certificate	194
Let's Encrypt	194
Client installation	194
Certificate generation	194
Tomcat configuration	195
Other configuration	195
Known issues	196
Windows uses a different protocol	196
ca-bundles certificates	196
Additional information	196
Configuring Tomcat timezone	197
Configuring Tomcat to work behind an Internet proxy	198
Linux	198
Windows	199
Step one	199
Step two	200
Change repository home	202
Understanding the contents of repository folders	202
Move entire repository location	202
Sample	203
Windows	203
Linux	203
Move repository folders individually	203
Sample	203
Windows	203
Linux	203
Troubleshooting	204
Windows mapped network disk is not working	204
Change default role names	205
Change the default roles	205
Sample of appContext.xml file	206
Tools	211
Enable extensions	212

Enable WebDAV	214
Linux	214
Windows	215
WebDrive	215
Windows Web Folders	216
Windows XP's "My Network Places"	217
Other tools	217
Mac	217
Mac OS X finder.	217
Troubleshooting	218
Slow WebDAV performance on Windows	218
Turn off file locking	218
LDAP configuration	219
OpenLDAP	219
Active Directory	219
Uncategorised examples shared by users	219
Tools	219
OpenLDAP example with URL base at login	220
LDAP Structure	220
OpenKM.xml	221
Configuration parameters	222
OpenLDAP example retrieving attributes by name	224
LDAP Structure	224
OpenKM.xml	225
Configuration parameters	226
Active Directory example with referral enabled	228
LDAP Structure	228
OpenKM.xml	229
Configuration parameters	230
Active Directory example with login based on filtering users by roles	232
LDAP structure	232
OpenKM.xml	233
Configuration parameters	234
Active Directory basic configuration	237
LDAP structure	237
OpenKM.xml	237
Configuration parameters	237
Active Directory advanced configuration	240
LDAP structure	240
OpenKM.xml	240
Configuration parameters	241
LDAP best practices for filtering users and roles	243
Example based on single LDAP group	243
Example based on ROLE_USER and ROLE_ADMIN group	243
ClearOS LDAP configuration	245
LDAP Structure	245
OpenKM.xml	245
Configuration parameters	246
Create a Certificate	248
LDAP troubleshooting	250
javax.naming.PartialResultException error	250
Can't connect to LDAP server	250
Slow login or not able to login	250
Force all users to be members of ROLE_USER	251

org.springframework.beans.factory.parsing.BeanDefinitionParsingException	252
LDAP: error code 8 - BindSimple: Transport encryption required	253
Ldap error: Caused by java.security.cert.CertificateException: No subject alternative DNS name matching	255
Log Configuration	254
Log4j based Log Configuration	255
Third-party application execution	255
Enable log authentication	255
OpenOffice and LibreOffice configuration	256
Linux	256
Debian & Ubuntu	256
CentOS	256
Windows	257
Check configuration	258
Security configuration	259
Differences between simple and recursive	259
Extending security	260
Configuration for security changes which take a long time	260
Cache configuration	262
Troubleshooting	263
Server configuration	264
Packet for query is too large (MySQL)	264
Not able to connect from other computer in the network	264
Error downloading document as PDF	264
IMAP SSL certificate error	264
OpenKM server has no direct Internet connection	265
Access denied installing OpenKM as service on Windows 7	265
Problems with character encodings	265
PermGen errors	265
LDAP Sizelimit exceeded	266
Behind IIS proxy you get error 404 or 404.15	266
IIS gets IOException: Internal Server Error while uploading file	268
Solution one	268
Solution two	268
Zoho fails with SSL connection	269
Sigar java.lang.UnsatisfiedLinkError Known issue in Windows	269
WARN net.sf.ehcache.hibernate.strategy.AbstractReadWriteEhcacheAccessStrategy	270
OpenOffice is not starting on Windows	270
Client configuration	271
ZIP filenames with wrong charset	271
Microsoft Office Addin disappears after Office restart	272
Error in Uploader / Scanner applet	272
Browser configuration	273
Firefox cause error when click on "Copy to clipboard"	273
Unable to run Scanner applet under Linux	273
Internet Explorer error "Document name is different"	273
Internet Explorer error downloading files in HTTPS / SSL connections	273
Google Chrome and Java applets	274
Best practices reporting issues	275
Getting the stack trace	275
Linux	275
Windows	275
Sample of stack trace capture	276
Administration guide	278
Configuration parameters	281
Edit a parameter	281

Delete a parameter	281
Backup configuration parameters	281
Restore configuration parameters	282
Check configuration	282
Recommended configuration parameters	283
Preview parameters	286
Linux	286
Windows	286
LDAP configuration parameters	288
Sample query	288
LDAP queries	288
Parameters	289
Sample configuration	291
User interface configuration parameters	294
Workflow configuration parameters	297
Notification configuration parameters	298
Substitutions	300
Document version numeration configuration	301
Version numeration schemas	301
Migration between versions	302
Password validation configuration	303
Sample configuration	304
Switch application to read-only mode	305
Security configuration parameters	306
Extensions configuration parameters	309
HTML Editor 3	309
DXF applet viewer	314
Zoho	314
OpenMeetings	315
DICOM preview	315
Extratab extension	315
Performance configuration parameters	317
Text extractor queue	317
OpenOffice LibreOffice conversion service	318
Antivirus checker	318
Re-indexing the whole Lucene repository	319
Execution timeout	319
Uploading bandwidth	320
Remote conversion service	320
Activity log actions	320
OpenOffice & LibreOffice configuration parameters	322
WebDAV configuration parameters	323
CSV configuration parameters	324
Other configuration parameters	325
Experimental configuration parameters	331
AutoCAD configuration parameters	332
Thesaurus configuration	333
Datastore	335
File system datastore	335
Database datastore	335
MIME types	336

Add new MIME type	336
Edit MIME Type	337
Delete MIME type	337
Backup MIME types	337
Restore MIME types	337
Statistics	338
Check the status of the Pending task queue	338
Check the status of the Text extraction queue	338
Scripting	340
Execute script as part of application starting process	340
Execute script as part of application stopping process	340
Scripting in Automation	340
Metadata	342
Edit metadata	343
Metadata XML definition	344
Basic XML skeleton	344
Metadata group definition	344
Basic example	345
Using file system DTD	346
Metadata fields	347
Metadata Checkbox field	349
Checkbox	349
Metadata Select field	351
Basic select example	352
Allow multiple choices	353
Options based on a metadata table	353
Suggested options	354
Metadata Separator field	356
Metadata Suggestbox field	357
Metadata Text field	360
Metadata TextArea field	361
Metadata Validator element	363
Basic example	364
Regular expression	364
Metadata Option element	365
Basic select with selected option	365
Metadata Input field	366
Manage users and roles	368
Manage users	369
Create a user	370
Edit a user	370
Delete a user	370
Change the user profile	370
Export the users	371
Enable twitter account	371
Manage user mail accounts	372
Create a new mail account	373
Edit a mail account	374
Delete a mail account	374
Browser emails in the mail server	374
Manage mail account filters	374
Create a mail account filter	376

Edit a mail account filter	376
Delete a mail account filter	376
Create a mail account rule	376
Edit a mail account rule	376
Delete a mail account rule	377
Manage roles	378
Create a role	378
Edit a role	378
Delete a role	379
Logged users	380
Real time user activity log	380
Message queue	381
Send a message	382
Edit a message	382
Delete a message	382
Delete all messages queue	382
Refresh message queue	382
Profiles	383
Create a new profile	383
Edit a profile	383
Delete a profile	383
Clone a profile	384
Profile General tab	385
General section	385
Misc section	385
Chat section	386
Wizard section	386
Pagination section	387
Profile Components tab	389
Stacks section	389
Dashboard section	389
Tab section	390
Tab document section	390
Tab folder section	391
Tab mail section	391
Profile Menu tab	393
Menu section	393
Menu bookmark section	393
Menu tool section	394
Menu help section	394
Menu file section	395
Menu edit section	396
Profile Other tab	399
Toolbar section	399
File browser section	400
Database query	402
Execute JDBC query	402
Execute Hibernate query	403
Execute Metadata query	403
Allowed sentences	403
Execute queries from sql file	404
Reports	405
Create a new report	405
Edit a report	405

Delete a report	405
Show report report parameters	406
Execute a report	406
Reports pack	407
Template reports	407
Authentication	407
Documents	407
Folders	410
Mails	411
Others	412
Activity log	413
Auth actions	413
Document actions	414
Folder actions	415
Mail actions	416
Repository actions	417
Admin actions	417
Misc actions	422
Workflow	424
Register workflow	424
Workflow forms definition	426
Basic XML skeleton	426
Workflow task form definition	426
Basic example	427
Using file system DTD	427
Workflow fields	429
Workflow Button field	431
Workflow Download field	433
Basic download example	433
Dynamic download example	434
Workflow Validator element	435
Basic example	436
Regular expression	436
Workflow Node element	438
Basic download example	438
Workflow Upload field	439
Basic upload example	440
Document update	440
Upload example with dynamic parameters	440
Workflow Print field	442
Basic print example	442
Dynamic print example	443
Workflow Checkbox field	444
Checkbox	444
Workflow Input field	445
Workflow Select field	447
Basic select example	448
Allow multiple choice	448
Options based on metadata table	448
Based on SQL query	450
Workflow Separator field	451
Workflow Suggestbox field	452

Workflow Text field	455
Workflow TextArea field	456
Workflow Option element	457
Basic select with selected option	457
Automation	458
Preliminar concepts.	458
Understanding Validators and Actions	459
Creating a rule	460
Editing a rule	460
Edit general parameters	460
Edit action and validation definitions	460
Deleting a rule	461
Built in Automation Validator plugins	462
Built in Automation Action plugins	464
Enable Automation validations	467
Almost databases	467
Oracle	467
PostgreSQL	467
Enable Automation actions	469
Almost databases	469
Oracle	469
PostgreSQL	470
Crontab	471
Create a new crontab job	473
Edit a crontab job	473
Execute a crontab job	473
Download a crontab job	473
Delete a crontab job	474
Built in crontab tasks	475
OMR templates	476
Add a new OMR template	476
Preliminars	477
Create a new template	478
Update asc training files	478
Update .fields training files	481
Check the template	483
Thesaurus	485
Build thesaurus	485
Languages	486
Create a new language	486
Edit a language	486
Delete a language	486
Edit language translations	487
Export a language translations	487
Import a language translations	487
Language translations	488
How to create a new translation	491
A sample of new language base on Oromo from Ethiopia	491
Repository import	492
Import	493
Repository export	494
Export	494
Understanding output format	494

Case metadata and history are not checked	494
Case only metadata is checked	495
Case only history is checked	496
Case metadata and history are checked	496
Utilities	497
Cache stats	498
Clean all cache	499
Refresh cache stats	499
Enable cache stats audit trail	499
Disable cache stats audit trail	499
Clean cache stats audit trail	500
Check email	501
Check email	501
Check text extraction	502
Check text extraction	502
Hibernate stats	503
Information provided by Hibernate stats	503
Enable Hibernate stats	503
Clear Hibernate stats	504
Disable Hibernate stats	504
List indexes	505
Search indexes	506
Enable show terms	507
Disable show terms	507
LogCat	508
Purge all logs	508
Download file as ZIP	508
Browser log file	508
Profiling stats	510
Enable profiling stats	511
Disable profiling stats	511
Clear profiling stats	511
Refresh profiling stats	511
List of detailed calls	511
Purge permissions	512
Purge	512
Rebuild indexes	513
Rebuild	513
Repository checker	514
Check repository	514
Repository view	515
Node folder selected	515
Node document selected	515
Node mail selected	516
System properties	517
CSS	519
Create a new CSS	519
Edit a CSS	519
Delete a CSS	519
Clone a CSS	519
Download a CSS	520
Samples	520
Large font sample	520
Medium font sample	520

Default font sample	520
User guide	522
Menu bar	523
Quick search box	523
ToolBar	523
Desktop	523
Navigation	523
Browser	524
Properties	524
Search	525
Dashboard	526
Administration	527
Status	527
Quick start	529
Loggin in	529
Create a taxonomy	529
Add documents	531
Edit documents	532
Search documents	534
Essential fundamentals	536
Basic concepts	536
Understanding taxonomy, keywords, categories and metadata groups.	536
Taxonomy	536
Keywords	537
Categories	537
Metadata	537
Plan your data organization	539
Good practice	540
Working in menu bar	541
File	541
Edit	544
Tools	546
Bookmark	549
Templates	549
Help	550
Working in toolbar	551
Working in desktop	555
Navigation	555
Browser	558
Pagination	562
Properties	563
Tabs on nodes of type document	563
Tabs on nodes of type folder	566
Tabs on nodes of type mail	568
Shortcuts	570
Working in search	572
Searches panel	572
Search panel	573
Operators and wildcards	573
Default search	573
Advanced search	574
Control panel	577
Results	578
Working in dashboard	580
User view	580
General view	581

Mail	582
News	582
Workflow	582
Workflow lists	582
Workflow detail	583
Keyword map	583
Toolbar	583
Browser	584
Keywords lists	584
Results controller	584
Messaging	585
Message type	585
Browser	586
Preview	586
Forum	586
Wiki	587
Working in status bar	589
OpenKM news alert	589
The connected user	589
User repository size	589
Chat	589
Alerts	589
Message panel	590
Bulk Uploader	591
Usage	591
HTML editor extension	592
Features	592
Edit HTML	592
Editor panel	593
Preview panel	594
Extratab extension	595
Configuration	595
Sample	595
Forum extension	597
Editor features	597
Create new topic	598
Create reply	598
Update post	599
Delete post	599
Create forum	599
Update forum	599
Delete forum	599
Wiki extension	600
Editor features	600
Creating a new wiki page	601
Update wiki page	601
Delete wiki page	602
Show history	602
Unlock wiki page	602
Stapling extension	603
Create new stapling group	603
Working with stapling group	603
Add new relations	603
Delete relations	603
Download nodes	603
Using OpenKM from external applications	605
Microsoft Office Add-in	605

Digital Signature Client	605
Scanner client	605
Desktop Importer tool	605
Microsoft Office Add-in	606
Configuring add-in	606
Editing documents from Microsoft Office	606
Importing mails from Microsoft Outlook	607
Add new translation	607
Know issues	608
Digital Signature Client	609
Configuring	609
Sign documents	609
Scanner client	610
Requirements	610
Configuring	610
Compression CCITT Group4	611
Setting destination folder	611
Setting scanner configuration	611
Scanning	611
Convert scanned images to PDF	611
Uploading	612
Add images to an existing PDF file	612
Remove pages from existing PDF file	613
Desktop importer tool	614
Images	614
Migration guide	615
Changelog	616
OpenKM 6.3 Community	616
v6.3.9	616
v6.3.8	616
v6.3.7	616
v6.3.6	616
v6.3.5 - Nexus	616
v6.3.4 - Tachikoma	617
v6.3.3 - Reborn	617
v6.3.2 - Relative	617
v6.3.1 - Pumpkin	618
v6.3.0 - Serpico	619
Tomcat update	620
How to switch between databases	622
The process of migration between two databases can be divided into several steps:	622
Generate users and roles SQL scripts	622
From HSQLDB	622
From MySQL	622
Generate database configuration parameters SQL scripts	622
Export Metadata	622
Export crontab	622
Export the repository	622
Installation of OpenKM	623
Import the repository	623
Execute scripts	624
Restore configuration parameters	624
Restore metadata	624
Restore crontab task	625
Data what cannot be migrated and might be set again	625
Migrating from 6.3.9 to 6.3.10	626

Preliminaries	626
Migration	626
Migrating from 6.3.8 to 6.3.9	627
Preliminaries	627
Migration	627
Migrating from 6.3.7 to 6.3.8	628
Preliminaries	628
Migration	628
Migrating from 6.3.6 to 6.3.7	629
Preliminaries	629
Migration	629
Update automation definition	629
MySQL	629
PostgreSQL	629
HQL	629
Custom automation actions or validations	629
All databases	630
Migrating from 6.3.5 to 6.3.6	631
Preliminaries	631
Migration	631
Migrating from 6.3.4 to 6.3.5	632
Preliminaries	632
Migration	632
Migrating from 6.3.3 to 6.3.4	633
Preliminaries	633
Migration	633
Migrating from 6.3.2 to 6.3.3	634
Preliminaries	634
Migration	634
Migrating from 6.3.1 to 6.3.2	636
Preliminaries	636
Migration	636
Migrating from 6.3.0 to 6.3.1	637
Preliminaries	637
Migration	637
Migrating from 6.2.6 to 6.3.0	638
Preliminaries	638
Migration	638
MySQL	638
PostgreSQL	638
HSQL	638
Development	640
OpenKM 6.3 community portable development environment	640
Javadoc	641
Configuring IDE from a clean Eclipse	642
Requirements	642
Software:	642
Hardware:	642
Download eclipse	642
Configure Java	642
Configure tomcat server	644
Maven support for Eclipse	650
Subversion support for Eclipse	650
Configuring Maven in your OS	652
Maven tips	652

Memory problems	652
Configure eclipse to use local Maven	653
Checkout source code from Git	654
Checkout source code from Subversion	660
Eclipse tips	663
Debug tomcat	663
Debug remote tomcat server	663
Eclipse configuration	663
Debug GWT frontend user interface with GWT Super Dev Mode	665
Enable source for debugging	665
Eclipse configuration	666
Debug GWT frontend user interface with GWT Shell	675
Enable source for debugging	675
Eclipse configuration	676
Eclipse know troubleshooting	683
Maven raise error 206 the filename or extension is too long	683
Development tips	684
Tomcat application deployment	684
Compile only GWT frontend module	684
Configure GWT build from eclipse	684
Compile only one GWT frontend permutation	686
Enable only firefox permutation	686
Enable only chrome permutation	687
Enable only IE permutation	687
Configure number of CPU's to be used for compiling GWT frontend	687
Deploying script sample	688
Profiling application	689
Edit Tomcat startup scripts	689
Authentication	689
Monitoring Remote JVM Over SSH	690
HPROF	691
Application version compatibility table	692
Creating a JAR file with Eclipse	693
Images	693
Creating plugins	697
Register a new plugin	698
Creating your own Automation Action	699
Methods description	699
Understanding env variable	701
Understanding params variable	701
Description of available ParamType values:	702
Description of available Src values:	702
Detecting recursion	702
Example	703
Creating your own Automation Validation	706
Methods description	706
Understanding env variable	708
Understanding params variable	708
Description of available ParamType values:	708
Description of available Src values:	709
Example	709
Creating templates	712
Creating OpenOffice and LibreOffice templates	713
Preliminaries	713
Creating the template	713

Date format	714
Dynamic tables	715
Creating PDF template	716
Preliminaries	716
Creating the template	717
Date format	717
Creating Text templates	719
Preliminaries	719
Creating the template	719
Date format	720
Creating your own Automation Action (deprecated)	721
Methods description	721
Understanding env variable	721
Understanding params variable	722
Preventing recursion	722
Register	722
Example	725
Creating your own Automation Validation (deprecated)	728
Methods description	728
Understanding env variable	728
Understanding params variable	729
Register	729
Example	731
Creating your own Crontab job	734
Understanding the crontab job execution context	734
Crontab job based on JAR file	734
Contab job based on BeanShell	735
Creating your own database tables	736
Create a new table definition	736
Example	736
Create a new table values	737
Example	738
Create a new sequence	739
Example	739
Code samples	739
Select a query	740
Update query	740
Delete query	740
Get next sequence value	740
Use your own database table from Java	742
Methods	742
Sample	742
Register database metadata table definition	742
Java classes	742
Advanced	744
Use your own database table from frontend (GWT)	745
Methods	745
Sample	745
Register database metadata table definition	745
Java classes	745
Advanced	748
Creating your own login module	749
PrincipalAdapter sample based on properties	751
Creating your own parameters	754
Sample	754
Creating your own password validator	755

Sample	755
Creating your own Principal Adapter	758
Methods description	760
Example of the Principal adapter implementation	761
Creating your own thesaurus	765
Creating your own Suggestion Analyzer	768
Methods description	768
Example of Suggestion Analyzer	768
Creating your own Text extractor	770
Methods description	771
Example of the Text extractor implementation	771
Creating your own reports	775
Introducing to reports based on SQL	776
Create datasource	776
Testing SQL sample report	777
Closer look into JRXML file	780
Query section	780
Fields section	781
Remote image section	781
Using fields in report	781
Introducing to reports based on Java	782
Testing Java sample report	782
Understanding the code sample	784
Using in report	785
Introducing to reports with filtering	787
Understanding filtering	787
Params XML file	787
Using fields in report	787
Java sample	788
SQL sample	790
Params form fields	792
Basic XML skeleton	792
Basic example	792
Using file system DTD	793
Params fields	794
Params Validator element	796
Basic example	797
Regular expression	797
Params Input field	798
Params Suggestbox field	799
Params Checkbox field	802
Checkbox	802
Params TextArea field	803
Params Select field	804
Basic select example	805
Allow multiple choice	805
Options based on metadata table	805
Based on SQL query	807
Params Separator field	808
Params Text field	809
Creating your own Version Number Adapter	810
Methods description	811
Example of the Version number adapter implementation	811

Database stuff	812
Webservices	813
CMIS	814
CMIS clients	814
RESTful	815
Sample usage	815
Folder samples	817
Document samples	817
Search samples	818
Security samples	818
Metadata samples	819
Notes samples	820
Swagger	821
How to accessing Swagger	821
Playing with Swagger	822
SOAP	826
JAVA samples	826
Change address dynamically from a request	827
Authentication	827
Metadata groups	827
.NET	828
Create a single webservice	828
Create all webservices in one library package	828
Suggested changes in generated classes	829
PHP	829
Remote method info	829
Authentication	829
List folders and documents	830
Create document	831
Create document and set a category	831
Search by content	832
Search with parameters	832
Set metadata group values	833
Exception handling	834
Proxy configuration	834
Extending frontend UI	835
Enable sample extensions	835
Frontend UI hello world sample	837
Description	837
Sample	837
Create a new GWT module	837
Add module into OpenKM frontend UI	837
Create new widget	837
Use widget into OpenKM	838
Results	838
Frontend UI extensions	840
Widget library	842
Tab workspace	842
Tab Document	842
Tab folder	843
Main menu	844
Toolbar button	844
Dashboard toolbar button	844
TabWorkspace Extension	846
Methods	846
Basic sample	846
Sample with iframe	846

WorkspaceHandlerExtension	849
Methods	849
Sample	849
WorkspaceComunicator	850
Methods	850
Sample	850
TabDocument Extension	851
Methods	851
Basic sample	851
DocumentHandlerExtension	852
Methods	852
Sample	852
TabDocumentComunicator	854
Methods	854
Sample	855
TabFolderExtension	856
Methods	856
Sample	856
FolderHandlerExtension	857
Methods	857
Sample	857
TabFolder Comunicator	858
Methods	858
Sample	858
TabMail Extension	859
Methods	859
Sample	859
MailHandlerExtension	860
Methods	860
Sample	860
TabMail Comunicator	861
Methods	861
Sample	861
ToolBarBox Extension	862
Methods	862
Sample	862
DashboardHandlerExtension	863
Methods	863
Sample	863
ToolBarComunicator	865
Methods	865
Sample	865
ToolBarButton Extension	866
Methods	866
Sample	866
ToolBarHandler Extension	868
Methods	868
Sample	868
MenuBarExtension	870
Contructor	870
Sample	870
MenuItemExtension	872

Constructors	872
Sample	872
General HandlerExtension stuff	874
NavigatorHandlerExtension	874
Methods	874
Sample	874
PropertyGroupHandlerExtension	874
Methods	874
Sample	874
WidgetHandlerExtension	875
Methods	875
Sample	875
General Communicator stuff	876
GeneralCommunicator	876
Methods	876
Sample	879
NavigatorCommunicator	879
Methods	879
Sample	880
FileBrowserCommunicator	880
Methods	881
Sample	881
UtilCommunicator	881
Methods	881
Sample	883
SearchCommunicator	883
Methods	883
Sample	884
ToolBarCommunicator	884
Methods	884
Sample	885
UI constants	886
UIMenuConstants	886
Sample	886
UIDesktopConstants	886
Sample	887
UIDockPanelConstants	887
Sample	887
UISearchConstants	887
UIFileUploadConstants	887
Frontend UI extensions stuff	888
Internatinalization	888
Sample	888
Registering RPC (Remote Process Call) service	889
Complete frontend extension sample	890
Description	890
Installation	891
Implementation details	892
index.jsp	892
menu.jsp	892
home.jsp	892
Sample 1	893
action.jsp	893
distributor.jsp	894
full_list.jsp	895
Screenshot of sample1 view	895
Sample 2	896
action.jsp	896
distributor.jsp	897
add_form.jsp	898
full_list.jsp	899
Screenshot of sample2 view	900

Using iframe for extending frontend UI	901
Exposed frontend JavaScript API	902
General functions	902
HTML Editor functions	903
Wiki functions	904
How to expose Java methods to JavaScript	904
Static method sample	904
Non Static method sample	904
Samples	906
Scripting samples	907
Script - Basic samples	909
Update repository statistics	909
Show jBPM mail configuration	909
Reset user document size	909
Refresh user items	909
Create some folders and set property group	909
Show number of documents, folders and size from a given path	910
Get path by UUID	910
Show all documents used in every running workflow	910
Script - Cognitive PDF/A	911
Images	912
Scanned files	912
Crontab compressimo Importer task	913
OpenKM log server output of the importer task execution	913
Imported files into OpenKM	914
PDF/A compliance	914
Search engine results by content	915
Script - Compact documents with size equals to 0	916
Script - Count all children subfolder	917
Script - Exporter	918
Script - Export repository folder to server as a ZIP file	919
Script - Folders deep	920
Images	921
Execution	921
Results	921
Script - Generate database script	922
Script - Import data from CSV file	923
Image	925
Register the metadata group	925
Repository view	925
Execution	926
Script results	926
Imported metadata	927
Script - Import reports from a folder	928
Script - Purge all folders into some parent	929
Script - Purge all users trashes	930
Script - Put documents from folder and descendants into pending text extraction queue	931
Script - Recursive character renaming	932
Script - Recursive repository traversal	933
Script - Recursive change security	934
Script - Security change at low level	936
Images	939

Initial security status	939
UUID of the folder	940
Execution	941
Log	942
Rebuild Lucene indexes	943
Security changed applied	943
Script samples for automation	945
Script for automation - Autonumeric	946
Images	947
Register metadata group	947
Register automation rule	947
Result after uploading a new document	949
Script for automation - Execute OMR	950
Images	950
Register automation rule	950
Templates definition	951
Result	951
Script for automation - Set unique document name based on metadata	953
Images	955
Register metadata group	955
Register automation rule	955
Results	957
Script for automation - Simple auto tagging	959
Images	960
Register metadata database definition	960
Register automation rule	961
Result	962
Crontab samples	964
Crontab sample - Basic file system document importer	965
Images	965
Register a crontab task	965
Files in the application server	966
Result	966
Crontab sample - Document importer with metadata values	967
Images	971
Register metadata definition	971
Create crontab task	972
Files in application server	972
Result	972
Mail notification	972
Imported files into the repository	973
Script - Correct security in personal nodes	974
PHP SOAP Webservices samples	975
PHP SOAP Webservices samples - Automatic import	976
PHP SOAP Webservices samples - Download a file	977
PHP SOAP Webservices samples - File upload	978
Crontab task - XML importer	980
Images	985
Create the jar file	985
Register crontab task	985
Register metadata definition	985
Logfile folder	986
Output folder	986
Result	987
Vault of stuff	988
Convert Exchange Legacy DN	989
Download document with direct link	991

Import document with bash script using REST	992
Import document with bash script using SOAP	993
Creating users from scripts	997
API description	998
OKMAuth	1000
Basics	1000
Methods	1000
login	1000
login	1001
logout	1002
grantUser	1002
revokeUser	1003
getGrantedUsers	1004
grantRole	1004
revokeRole	1005
getGrantedRoles	1005
getUsers	1006
getRoles	1006
getUsersByRole	1007
getRolesByUser	1007
getMail	1008
getName	1008
changeSecurity	1009
createUser	1009
deleteUser	1010
updateUser	1010
createRole	1011
deleteRole	1011
updateRole	1012
assignRole	1012
removeRole	1013
OKMBookmark	1014
Basics	1014
Methods	1014
add	1014
get	1015
getAll	1015
remove	1016
rename	1016
OKMDashboard	1018
Basics	1018
Methods	1018
getUserCheckedOutDocuments	1018
getUserLastModifiedDocuments	1018
getUserLockedDocuments	1019
getUserSubscribedDocuments	1019
getUserSubscribedFolders	1020
getUserLastUploadedDocuments	1021
getUserLastDownloadedDocuments	1021
getUserLastImportedMails	1022
getUserLastImportedMailAttachments	1022
getUserDocumentsSize	1023
getUserSearchs	1023
find	1024
getLastWeekTopDownloadedDocuments	1025
getLastMonthTopDownloadedDocuments	1025
getLastWeekTopModifiedDocuments	1026
getLastMonthTopModifiedDocuments	1026
getLastModifiedDocuments	1027
getLastUploadedDocuments	1027
visiteNode	1028
OKMDocument	1030
Basics	1030

Methods	1031
create	1031
createSimple	1031
delete	1032
getProperties	1033
getContent	1033
getContentByVersion	1034
getChilds	1035
getChildren	1035
rename	1036
setProperties	1036
checkout	1037
cancelCheckout	1038
forceCancelCheckout	1038
isCheckedOut	1039
checkin	1039
checkin	1040
isCheckedOut	1041
getVersionHistory	1042
lock	1042
unlock	1043
forceUnlock	1043
isLocked	1044
getLockInfo	1044
purge	1045
move	1045
copy	1046
extendedCopy	1047
restoreVersion	1047
purgeVersionHistory	1048
getVersionHistorySize	1048
isValid	1049
getPath	1050
OKMFolder	1051
Basics	1051
Methods	1051
create	1051
createSimple	1052
getProperties	1052
delete	1053
purge	1053
rename	1054
move	1055
copy	1055
extendedCopy	1056
extendedCopy	1057
getChilds	1058
getChildren	1058
getContentInfo	1059
isValid	1059
getPath	1060
createMissingFolders	1060
OKMMail	1062
Basics	1062
Methods	1062
create	1062
getProperties	1064
delete	1065
purge	1065
rename	1066
move	1066
copy	1067
extendedCopy	1067
getChilds	1068
getChildren	1069
isValid	1070

getPath	1070
importEml	1070
importMsg	1071
OKMNote	1073
Basics	1073
Methods	1073
addNote	1073
get	1074
delete	1074
set	1075
list	1075
OKMNotification	1077
Basics	1077
Methods	1077
subscribe	1077
unsubscribe	1078
getSubscriptors	1078
notify	1079
proposedSubscription	1079
OKMProperty	1081
Basics	1081
Methods	1081
addCategory	1081
removeCategory	1082
addKeyword	1082
removeKeyword	1083
setEncryption	1084
unsetEncryption	1084
setSigned	1085
OKMPropertyGroup	1087
Basics	1087
Methods	1087
addGroup	1087
removeGroup	1088
getGroups	1089
getAllGroups	1089
getProperties	1090
getPropertiesSimple	1090
setProperties	1091
setPropertySimple	1092
setPropertiesSimple	1093
getPropertyGroupForm	1095
hasGroup	1095
getSuggestions	1096
registerDefinition	1097
OKMRepository	1099
Methods	1099
getRootFolder	1099
getTrashFolder	1099
getTrashFolderBase	1100
getTemplatesFolder	1100
getPersonalFolder	1101
getPersonalFolderBase	1102
getMailFolder	1102
getMailFolderBase	1103
getThesaurusFolder	1103
getCategoriesFolder	1104
purgeTrash	1104
getUpdateMessage	1105
getRepositoryUuid	1105
hasNode	1106
getNodePath	1106
getNodeUuid	1107
getAppVersion	1107

copyAttributes	1108
registerPropertyGroups	1109
OKMSearch	1110
Basics	1110
Methods	1113
findByContent	1113
findByName	1113
findByKeywords	1114
find	1115
findPaginated	1115
saveSearch	1116
updateSearch	1117
getSearch	1118
getAllSearchs	1118
deleteSearch	1119
getKeywordMap	1119
getCategorizedDocuments	1120
getCategorizedFolders	1121
getCategorizedMails	1121
getDocumentsByKeyword	1122
getFoldersByKeyword	1122
getMailsByKeyword	1123
getDocumentsByPropertyValue	1123
getFoldersByPropertyValue	1124
getMailsByPropertyValue	1124
findSimpleQuery	1125
findSimpleQueryPaginated	1126
findMoreLikeThis	1127
OKMStats	1128
Basics	1128
Methods	1128
getDocumentsByContext	1128
getDocumentsSizeByContext	1128
getFoldersByContext	1129
getMailsByContext	1129
OKMUserConfig	1131
Methods	1131
setHome	1131
getConfig	1131
OKMWorkflow	1133
Methods	1133
registerProcessDefinition	1133
deleteProcessDefinition	1133
getProcessDefinitionImage	1134
getProcessDefinitionForms	1135
runProcessDefinition	1135
sendProcessInstanceSignal	1136
endProcessInstance	1137
deleteProcessInstance	1138
findProcessInstances	1138
findAllProcessDefinitions	1139
findLatestProcessDefinitions	1139
findLastProcessDefinition	1140
findAllProcessDefinitionVersions	1141
getProcessInstance	1141
suspendProcessInstance	1142
resumeProcessInstance	1142
addProcessInstanceVariable	1143
deleteProcessInstanceVariable	1143
findUserTaskInstances	1144
findPooledTaskInstances	1145
findTaskInstances	1145
setTaskInstanceValues	1146
addTaskInstanceComment	1146
getTaskInstance	1147

setTaskInstanceActorId	1148
addTaskInstanceVariable	1148
deleteTaskInstanceVariable	1149
startTaskInstance	1149
endTaskInstance	1150
suspendTaskInstance	1151
resumeTaskInstance	1151
getToken	1152
addTokenComment	1152
suspendToken	1153
resumeToken	1153
sendTokenSignal	1154
setTokenNode	1154
endToken	1155
Commons classes	1157
ArchiveUtils	1159
Methods	1159
createZip	1159
createJar	1159
DbSessionManager	1161
Methods	1161
getSystemToken	1161
ExecutionUtils	1162
Methods	1162
runScript	1162
runJar	1162
runCmd	1163
FileLogger	1164
Methods	1164
info	1164
warn	1164
error	1165
FileUtils	1166
Methods	1166
getFileName	1166
getFileExtension	1166
FormatUtil	1168
Methods	1168
formatSize	1168
parseSize	1168
formatDate	1169
ISO8601	1170
Methods	1170
parseBasic	1170
formatBasic	1170
LegacyDAO	1172
Methods	1172
executeSQL	1172
executeHQL	1172
MailUtils	1174
Methods	1174
sendMessage	1174
sendMessage	1174
sendDocuments	1175
PathUtils	1177
Methods	1177
getParent	1177
getName	1177
getContext	1178
PDFUtils	1179

Methods	1179
merge	1179
PrincipalUtils	1181
Methods	1181
getUser	1181
StackTraceUtils	1182
Methods	1182
isCallingMe	1182
toString	1182
TemplateUtils	1184
Methods	1184
templateExists	1184
replace	1184
WorkflowUtils	1186
Methods	1186
findProcessInstancesByNode	1186
Database description	1187
Node types	1187
SQL example queries	1187
Mark documents to being indexed again	1188
Node version	1188
SQL example queries	1189
Node categories and keywords	1189
SQL example queries	1190
Node properties	1190
SQL example queries	1190

OpenKM 6.3 - CE



OpenKM Community version is licensed under terms of [GNU General Public Licence](#).

OpenKM is an Electronic Document and Record Management System (EDRMS). There's a lot of literature about document management terms like: DMS, EDRMS or CMS usually more influenced by marketing rules rather than objective reasons.

The differences between DMS, CMS, and EDRMS in most cases are subjective or not relevant from the Document Management point of view. Lately, it seems the term CMS has gotten some marketing success and actually is one of the most common used. However, we prefer the term **EDRMS**, perhaps with lesser success on the marketing world but from our point of view more strict and representative.

OpenKM helps you to:

- Day-to-day data capture.
- Storage.
- Modify and share physical and/or digital documents within the organization.
- Apply retention periods to stored items.
- Identify the owner of each record series and documents.
- Set a chain of custody and proper audit trail.
- When needed apply legal records holds.
- Managing records disposition.
- Preserving records through their life cycle.

Some of the OpenKM goals are:

- Increase **efficiency**. Faster documents approval, reduced manual data entry, automated recurring tasks among others.
- Set **compliance** levels. The system helps the organization to avoid penalties when regulators, auditors and other government authorities summons them for inspections.
- Preserve **historical content**. Follow the status of the document (determined by the different phases of the lifecycle).
- **Consolidate** the company knowledge on a single access point.



The OpenKM technical staff maximum is "**Try finding out the easy solution for each problem. If you cannot find it, try again.**"

Although we've tried to give our powerful application what it needs to solve problems with a minimalist solution, sometimes its not possible. OpenKM can solve day-to-day complex scenarios. It is not a closed application and has a lot of features that can be used and configured on several ways.

Please take your time to read the documentation, it is time well spent. If you have some doubt contact with us. Usually a problem have several ways to be resolved.

Your suggestions are always welcomed, we try creating a better application to solve real life problems in a more efficient way. Your suggestions help us growing the knowledge of EDRMS solutions and how it can be applied to real case scenarios.

Hardware and software requirements

This section presents information about the environment needed to run an OpenKM server, listing the required and optional software and providing some facts about their configuration.



This sections aims at helping you to validate or define the production environment. However, the check-list is not exhaustive and needs to be completed with the user's experience.

Requirements

OpenKM can be installed on any operating system (Linux, Windows, Mac OS X,) since it is a multi-platform application.



A Linux host is a good choice because this OS handles heavy I/O applications, it works with OpenKM quite well, and it is conveniently priced.

The only requirement to run the OpenKM is to have **Java 8** (also called Java 1.8) installed.



Make sure you're installing OpenJDK 1.8 if you install JRE the application will not run correctly.

Checking the Java version

1. Open a terminal:
 - On Linux or Mac OS X: **open a terminal**.
 - On Windows: press "windows" key + r, type **cmd** (or command) in the Run window and press "OK" or open the "Prompt command" from "Start > Programs > Accessories" menu.
2. Type **java -version** and press **Enter**.

If Java is correctly installed on your computer, the name and version of the Java virtual machine is displayed:

```
Java version "1.8.0_31"  
Java(TM) SE Runtime Environment (build 1.8.0_31-b13)  
Java HotSpot(TM) 64-Bit Server VM (build 25.31-b07, mixed mode)
```

If Java is not installed on your computer, it will fail to display the Java version. Then, you will need to install Java (see below).

If Java is installed but not included in the PATH, it will fail to find the Java command. Then, you will need to add \$JAVA_HOME/bin/ in your PATH (see: [How do I set or change the PATH system variable?](#)).

Installing Java

For Windows users

If the required version of Java is not installed on your computer:

1. Download it from the Oracle website ([Java 8](#)) and choose the appropriate platform for your hardware and Windows version.
2. Run the downloaded .exe file and follow the instructions displayed.

For Linux users

You can download Java 8 from [Oracle website](#) (choose the right architecture). You can find RPM and TGZ packages to install.

If you use Ubuntu the following is a better method:

```
$ sudo add-apt-repository ppa:webupd8team/java
$ sudo apt-get update
$ sudo apt-get install oracle-java8-installer
```

This way your OpenJDK will be upgraded every time a new release is available.

In case you have another OpenJDK installed (like GCJ), you can select the right one this way:

```
$ sudo update-alternatives --config java
```

For Mac OS X Users

Java packages and instructions for installation are available from the Oracle website:

- [Download](#)
- [Instructions](#)

Recommendations

Hardware configuration



OpenKM runs quite well on any computer with a common hardware configuration. For a better performance we recommend at least 4 GB RAM (have at least ≥ 2 GB of plenty RAM) and a fast hard disk like SATA or better.

A dual-core Intel based CPU with 3.20 GHz should be fine for a reduced number of users (30-50 concurrent users).

Be sure to have enough disk space for the document repository and its versions.

A Linux host is a good choice because this OS handles heavy I/O applications like this quite well, and it is not expensive.

It is highly recommended 64 bits architecture.



We suggest having a modern server with good performance.

Although OpenKM can be installed on a server with other applications, the best scenario is a dedicated server. Keep in mind that on a shared scenario other applications can have a negative effect on OpenKM.

For more information about hardware requirements consult with your sales & marketing representatives as they should provide you a hardware requirements estimation.

Try giving them as much detailed information as possible for a more accurate estimation:

- Estimated number of concurrent users.
- Type of users (% of reader and % of writers).
- Estimated first year number of files.
- Estimated number of files growing per year.
- Estimated document size average.
- Most common document types (MS Office, OpenOffice, PDF, etc.).



You can use your older history of documents and folders to get some of these values.

Browser support

OpenKM user interface is based on GWT framework, than means the supported browsers are determinated by GWT browser support.

Actual browsers supported:

- Firefox
- Internet Explorer 8, 9, 10, 11
- Safari 5, 6
- Chromium and Goggle Chrome
- Opera latest version

More information at [GWT Browsers and Servers](#).



We encourage your browser to be updated to the latest version.

We recommend Firefox and Chrome rather than IE, for several reasons; best performance, less security holes, follow the standards.

Operating systems

Linux (Known working configurations):

- Ubuntu
- Debian
- CentOS

- Red Hat Linux



Red Hat Linux is a more difficult scenario, because some packages aren't by default in the Red Hat official repositories.



Ubuntu is the preferred Linux choice. We run the tests against Linux distribution and we build the packages for it. However, any recent Linux distribution should be usable as long as:

- OpenJDK 8 is available.
- You can adapt the provided Linux scripts.
- [Thirdparty software requirements](#)

Windows (Known working configurations):

- For development: Windows 7, Windows 8
- For deployment: Windows Server 2003, Windows Server 2008, Windows Server 2012

Databases

Known working configurations:

- MySQL
- PostgreSQL
- Oracle
- SQL server



The application internally uses Hibernate, in fact you should get it running from any of the supported Hibernate dialects, for more information read [Hibernate: SQL Dialects](#).

JVM

Although OpenKM can run with other JDK we recommend the use of OpenJDK 8 (64 bits architecture is highly recommended).

LDAP

- Open LDAP
- Microsoft Active Directory

Third-party software requirements

OpenKM takes advantage of using third-party software for some features. Some of this software needs to be installed on the server, others are optional.

Software	Usage	Additional information	Required
Openoffice or Libreoffice	<ul style="list-style-type: none"> • Converts documents to pdf. • Needed for preview feature. 		Yes
Imagemagick	<ul style="list-style-type: none"> • Transforms images. • Generates thumbnails. • Needed for preview feature. 	<p>Usually comes by default installed in Linux OS.</p> <p>In Windows in most cases it is not necessary to install it, because it comes embedded by default with OpenKM bundle installation.</p>	Yes
SWFTools	<ul style="list-style-type: none"> • Converts pdf document to swf. • Needed for preview feature. 	<p>On most cases it is not necessary to install it, because it comes embedded by default with OpenKM bundle installation.</p>	Yes
GhostScript	<ul style="list-style-type: none"> • Interpreter for the PostScript language and for PDF. • Needed by Imagemagick. 	Usually comes by default installed on Linux OS.	Optional
Pdftohtml	<ul style="list-style-type: none"> • Converts pdf document to html. • Needed for preview feature. 		Optional

Tesseract or cuneiform	<ul style="list-style-type: none">• OCR engine.• Used for the text extraction feature.	Required when application must processes images to extract text (OCR).	Optional
Ffmpeg	<ul style="list-style-type: none">• Converts and streams audio and video		Optional
Clamav or equivalent	<ul style="list-style-type: none">• Antivirus	Required when antivirus feature is enabled.	Optional

Installation

OpenKM can be installed on several Operating Systems and can be configured to use different databases. Because it's implemented in Java, the only software requirement is having a Java Virtual Machine compiled for your platform. But OpenKM also depends on third-party applications to be fully configured. For example, it needs a working OpenOffice or LibreOffice installation to be able of preview documents. Or you may want to extract text from your scanned images, and for this an OCR engine is needed.

Before installing you need to prepare the server. Depending on the Operating System you should follow some tips. When you install the Operating System, keep on mind that OpenKM does not need a graphical user desktop. So, in Linux systems, you can avoid installing these packages and save some memory. Another important advice is to do a minimal Operating System installation. We only need the minimum to access the server. In case of Linux server, we prefer to access by SSH because is fast and secure. Please, don't install Java or Tomcat or any other software unless you have already asked us.

This is a checklist of information we should know before installing:

- How many document do you want to store?
- How many user will access OpenKM?
- How many concurrent users?
- How much memory does the server have?
- How many CPUs does the server have?
- How much free disk is available in the server?
- Which Operating System is installed?
- Is the installed Operating System the 64 bits version?

In case of Linux server there are some additional questions:

- Which distro (name and version)?
- Does the server have a GUI (X-Windows) installed?
- It is a clean and minimal installation (only SSH) or have you installed another software?
- Is this server used for other applications?
- Is the database already installed?

In this section you will learn how to install OpenKM in your server and how to match most common third-party applications requirements.

OpenKM has been successfully installed on:

OS	Description

Linux	Ubuntu Server
	Linux Mint
	Debian
	CentOS
	Red Hat
	SuSE
Windows	Windows Server 2012 R2
	Windows Server 2008 R2
	Windows Server 2003 R2
	Windows 8
	Windows 7

Installing on Ubuntu and Debian

Preliminaries

Check disk size

```
$ df -h
```

Create user named openkm

```
$ sudo adduser openkm
```



You can easily generate random passwords at:

- <https://www.random.org/passwords>.
- <https://www.safetydetectives.com/password-meter/>

Check if partners channels are enabled

```
$ cat /etc/apt/sources.list
```

In case they are not enabled, enable them.

```
$ sudo vim /etc/apt/sources.list  
$ sudo apt-get update
```

Check server configuration

```
$ wget -Nc smxi.org/inxi  
$ sudo chmod +x inxi  
$ sudo ./inxi -F
```

Increase ulimit

```
$ sudo vim /etc/pam.d/su
```

uncomment the following line

```
#session    required    pam_limits.so
```

to

```
session    required    pam_limits.so
```



```
$ sudo vim /etc/security/limits.conf
```

and add the following lines to the end of the file (before the line # End of file)

```
* soft nofile 6084
* hard nofile 6084
```



Now reboot the server, execute **ulimit -n** and check if the open files limits are now updated.

Checking Java version

```
$ java -version
```

If Java is correctly installed on your computer, the name and version of the Java virtual machine is displayed:

```
openjdk version "1.8.0_212"
OpenJDK Runtime Environment (build 1.8.0_212-8u212-b03-0ubuntu1.18.04.1-b03)
OpenJDK 64-Bit Server VM (build 25.212-b03, mixed mode)
```



The Java version may be different depending on your installed one, but we always recommend the latest 1.8 release.

Install Java



Perform these steps only if Java 8 is not already installed on your server.

In case of Ubuntu

```
$ sudo apt-get update
$ sudo apt-get install openjdk-8-jdk
```

In case of Debian

```
$ su -
$ apt-get update
$ apt-get install openjdk-8-jdk
$ exit
```

Set OpenJDK 1.8 as your default Java version



Perform this only if Java 8 is not your default Java version.

```
$ sudo update-alternatives --config java
```

Installing database

Install MySQL if you do not have it already installed:

```
$ sudo apt-get install mysql-server
```

When you install MySQL, the installer will ask you for a password for the root user. Keep this password handy.

Also you can install MariaDB, which replaces MySQL in latest versions:

```
$ sudo apt-get install mariadb-server
```

Change MySQL root password

Do it only if it's necessary to change MySQL root password.

Method 1

```
$ /usr/bin/mysqladmin -u root -h localhost password 'password'
```

Method 2

```
$ sudo /etc/init.d/mysql stop
$ sudo mysqld --skip-grant-tables &
$ mysql -u root mysql
> UPDATE user SET Password=PASSWORD('YOURNEWPASSWORD') WHERE User='root';
> FLUSH PRIVILEGES;
> exit;
```



MySQL in this case has not been started at service. The process must be killed to stop MySQL and start it again as a service.

Method 3

```
$ mysql
> UPDATE mysql.user SET Password=PASSWORD('MyNewPass') WHERE User='root';
> flush privileges;
```

Method 4

```
$ sudo /etc/init.d/mysql stop
$ mysqld_safe --skip-grant-tables &
```

```
$ mysql -u root mysql
> UPDATE user SET Password=PASSWORD('YOURNEWPASSWORD') WHERE User='root';
> FLUSH PRIVILEGES;
> exit;
```



MySQL in this case has not been started at service. The process must be killed to stop MySQL and start it again as a service

More information at [MySQL : Reseting permissions](#).

Check InnoDB is default MySQL engine

```
$ mysql -h localhost -u root -p
> show engines;
```



It should looks something like this:

| **InnoDB** | **DEFAULT** | Supports transactions, row-level locking, and foreign keys | YES | YES | YES |

If it not default, modify `/etc/mysql/my.cnf`

```
$ vim /etc/mysql/my.cnf
```

and under `[mysqld]` add

```
default-storage-engine = innodb
```



After changing default MySQL engine the MySQL service must be restarted to take effect.

Create database

```
CREATE DATABASE okmdb DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_bin;
CREATE USER openkm@localhost IDENTIFIED BY 'password';
GRANT ALL ON okmdb.* TO openkm@localhost WITH GRANT OPTION;
```

Installing OpenKM and Tomcat bundle

```
$ cd /home/openkm
$ unzip openkm-6.3.2-community-tomcat-bundle.zip
```



The tomcat version numeration name can be altered in order to upgrade to a major version.

Configuring Tomcat as a service



For security reasons you shouldn't run Tomcat as **root**. It is better to use the user named openkm.

Create a file with the script:

```
$ sudo vim /etc/init.d/tomcat
```

```
#!/bin/sh

### BEGIN INIT INFO
# Provides:          tomcat
# Required-Start:    $remote_fs $syslog
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start and stop Apache Tomcat
# Description:       Enable Apache Tomcat service provided by daemon.
### END INIT INFO

ECHO=/bin/echo
TEST=/usr/bin/test
TOMCAT_USER=openkm
TOMCAT_HOME=/home/openkm/tomcat
TOMCAT_START_SCRIPT=$TOMCAT_HOME/bin/startup.sh
TOMCAT_STOP_SCRIPT=$TOMCAT_HOME/bin/shutdown.sh

$TEST -x $TOMCAT_START_SCRIPT || exit 0
$TEST -x $TOMCAT_STOP_SCRIPT || exit 0

start() {
    $ECHO -n "Starting Tomcat"
    su - $TOMCAT_USER -c "$TOMCAT_START_SCRIPT &"
    $ECHO "."
}

stop() {
    $ECHO -n "Stopping Tomcat"
    su - $TOMCAT_USER -c "$TOMCAT_STOP_SCRIPT 60 -force &"
    while [ "$(ps -fu $TOMCAT_USER | grep java | grep tomcat | wc -l)" -gt "0" ]; do
        sleep 5; $ECHO -n "."
    done
    $ECHO "."
}

case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        stop
        sleep 30
        start
        ;;
    *)
        $ECHO "Usage: tomcat {start|stop|restart}"
        exit 1
esac

exit 0
```

```
exit 0
```



The script must be changed if you use other user to start the service or your \$TOMCAT_HOME is not /home/openkm/tomcat

Make it executable:

```
$ sudo chmod 755 /etc/init.d/tomcat
```

Update the run-levels:

```
$ sudo update-rc.d tomcat defaults
```

Check the service

Start the service:

```
$ sudo service tomcat start
```

Stop the service:

```
$ sudo service tomcat stop
```

Installing third-party software

Software	Required	Command
LibreOffice or OpenOffice	Yes	<pre>\$ apt-get install libreoffice</pre>
Tesseract	No	<pre>\$ apt-get install tesseract-ocr tesseract-ocr-eng</pre> <div> <p>Depending on your locale you would like to install other language file.</p> </div>
Clamav	No	<pre>\$ apt-get install clamav</pre>
Imagemagick	Yes	<pre>\$ apt-get install imagemagick</pre>
Ghostscript	Yes	<pre>\$ apt-get install ghostscript</pre>

Start application

Check OpenKM.cfg parameters

```
$ vim /home/openkm/tomcat/OpenKM.cfg
```

Your configuration should be something like this:

```
# OpenKM Hibernate configuration values
hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect
hibernate.hbm2ddl=create
```



Ensure **hibernate.dialect=org.hibernate.dialect.MySQL5Dialect** is enabled if your database is MySQL.

First time application start:

```
$ service tomcat start
```

To see the startup process can execute the command:

```
$ tail -f /home/openkm/tomcat/log/catalina.log
```

When application is started you'll see in log file the lines:

```
2015-07-04 18:28:10,680 [main] INFO org.apache.coyote.http11.Http11Protocol - Starti
2015-07-04 18:28:10,688 [main] INFO org.apache.coyote.ajp.AjpProtocol - Starting Pro
2015-07-04 18:28:10,692 [main] INFO org.apache.catalina.startup.Catalina - Server st
```

The application will be accessible at http://YOUR_IP:8080/OpenKM by user: **okmAdmin** with password: **admin**. (Do not forget to change it !). For more information about [Manage users and roles](#).

Configure default extensions:

Go to Administration > [Database query](#) and execute:

```
INSERT INTO OKM_EXTENSION (EXT_UUID, EXT_NAME) VALUES ('808e7a42-2e73-470c-ba23-e4c9d
INSERT INTO OKM_EXTENSION (EXT_UUID, EXT_NAME) VALUES ('58392af6-2131-413b-b188-1851a
INSERT INTO OKM_PROFILE_MSC_EXTENSION (PEX_ID, PEX_EXTENSION) VALUES (1, '808e7a42-2e
INSERT INTO OKM_PROFILE_MSC_EXTENSION (PEX_ID, PEX_EXTENSION) VALUES (1, '58392af6-21
```

Other common additional configurations setup

OpenKM comes with a lot of configuration parameters. Please make sure to review: [Recommended configuration parameters](#).

- [Configuring Apache HTTP Reverse-Proxy](#)
- [Configuring mail](#)

- [Configuring Tomcat memory utilization](#)
- [Configuring Tomcat port](#)
- [Configuring Tomcat SSL/TLS](#)
- [Configuring Tomcat timezone](#)

Restart installation from scratch

The OpenKM first startup does some background configurations like filesystem folder creation, changes in the OpenKM.cfg what only happens during the first startup.

Sometimes maybe some mistake in OpenKM configuration, for example, wrong database connection or similar issues during the first startup what raise an application failure. To restart the installation from scratch we suggest following the next steps.

1. Ensure OpenKM is stopped.
2. Edit the OpenKM.cfg and set the parameter **hibernate.hbm2ddl=create**
3. In case of issue with database connection, we suggest checking the connection from the OpenKM server. Remember some databases require enabling the user to gain connection from a server (specific IP, range of IP, any server).
4. Delete the folder named "**repository**"
5. Start OpenKM.

Installing on RedHat and CentOS



If you decide installing Redhat, take in consideration you should have a licensed Redhat version, otherwise the repositories for installing software are locked.

Does not have much sense installing Redhat without a license subscription in the same direction does not have much sense enable CentOS repos in a Redhat, if this is your way better installing CentOS at the beginning.

Preliminaries

Check disk size

```
$ df -h
```

Create a user named openkm

```
$ sudo adduser openkm
```



You can easily generate random passwords at <https://www.random.org/passwords>.

Check server configuration

```
$ wget -Nc smxi.org/inxi
```

```
$ sudo chmod +x inxi
```

```
$ sudo ./inxi -F
```



If it doesn't work, try installing this package:

```
$ sudo yum install pciutils
```

Increase ulimit

```
$ sudo vim /etc/security/limits.conf
```

add the following lines to the end of the file (before the line # End of file)

```
* soft nofile 6084
* hard nofile 6084
```



Execute **ulimit -n** and see the open files limits are now updated.

Checking Java version

```
$ java -version
```

If Java is correctly installed on your computer, the name and version of the Java virtual machine is displayed:

```
openjdk version "1.8.0_212"  
OpenJDK Runtime Environment (build 1.8.0_212-b04)  
OpenJDK 64-Bit Server VM (build 25.212-b04, mixed mode)
```



The Java version numeration name can be altered in order to be upgraded to major version.

Install Java



Perform these steps only if Java 8 is not already installed on your server.

```
$ sudo yum install java-1.8.0-openjdk-devel
```



The Java version may be different depending on your installed one, but we always recommend the latest 1.8 release.

Set OpenJDK 1.8 as your default Java version



Perform this only if Java 8 is not your default Java version.

```
$ alternatives --config java
```

Installing database

Install MySQL if you do not have it already installed:

```
$ sudo yum install mysql-server mysql
```

Configure it as service:

```
$ sudo chkconfig --levels 235 mysqld on
```



On latest Centos the MySQL package has been replaced in the repositories by MariaDB.

Install MariaDB:

```
$ sudo yum install mariadb-server
```

Configure as service:

```
$ sudo systemctl enable mariadb
```



If you get some trouble on the first database startup because it can not create files into tmp folder do:

Edit the /etc/selinux/config file:

```
$ sudo vim /etc/selinux/config
```

And apply this change to:

```
'SELINUX=disabled'
```

To take effect immediately, execute from command line:

```
$ sudo setenforce 0
```

Change MySQL root password

Do it only If it's necessary to change the MySQL root password.

Method 1

```
$ /usr/bin/mysqladmin -u root -h localhost password 'password'
```

Method 2

```
$ sudo /etc/init.d/mysql stop
$ sudo mysqld --skip-grant-tables &
$ mysql -u root mysql
> UPDATE user SET Password=PASSWORD('YOURNEWPASSWORD') WHERE User='root';
> FLUSH PRIVILEGES;
> exit;
```



In this case MySQL has not been started as a service. The process must be killed to stop MySQL and start it again as service.

Method 3

```
$ mysql
> UPDATE mysql.user SET Password=PASSWORD('MyNewPass') WHERE User='root';
> flush privileges;
```

Method 4

```
$ sudo /etc/init.d/mysql stop
$ mysqld_safe --skip-grant-tables &
$ mysql -u root mysql
> UPDATE user SET Password=PASSWORD('YOURNEWPASSWORD') WHERE User='root';
> FLUSH PRIVILEGES;
> exit;
```



In this case MySQL has not been started as service. The process must be killed to stop MySQL and start it again as service.

More information at [MySQL : Reseting permissions](#).

Check InnoDB is default MySQL engine

```
$ mysql -h localhost -u root -p
> show engines;
```



It should show something like this:

| **InnoDB** | **DEFAULT** | Supports transactions, row-level locking, and foreign keys | YES | YES | YES |

If it not default, modify `/etc/mysql/my.cnf`

```
$ vim /etc/mysql/my.cnf
```

and under `[mysqld]` add

```
default-storage-engine = innodb
```



After changing the default MySQL engine the MySQL service must be restarted to take effect.

Create database

```
CREATE DATABASE okmdb DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_bin;
CREATE USER openkm@localhost IDENTIFIED BY 'password';
GRANT ALL ON okmdb.* TO openkm@localhost WITH GRANT OPTION;
```

Installing OpenKM and Tomcat bundle

```
$ cd /home/openkm
$ unzip openkm-6.3.2-community-tomcat-bundle.zip
```



The tomcat version numeration name can be altered in order to be upgraded to major version.

Configuring Tomcat as a service



For security reasons you shouldn't run Tomcat as **root**. It is better to use the user named openkm.

Create a file with the script:

```
$ sudo vim /etc/init.d/tomcat
```

```
#!/bin/sh

### BEGIN INIT INFO
# Provides:          tomcat
# Required-Start:    $remote_fs $syslog
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start and stop Apache Tomcat
# Description:       Enable Apache Tomcat service provided by daemon.
### END INIT INFO

ECHO=/bin/echo
TEST=/usr/bin/test
TOMCAT_USER=openkm
TOMCAT_HOME=/home/openkm/tomcat
TOMCAT_START_SCRIPT=$TOMCAT_HOME/bin/startup.sh
TOMCAT_STOP_SCRIPT=$TOMCAT_HOME/bin/shutdown.sh

$TEST -x $TOMCAT_START_SCRIPT || exit 0
$TEST -x $TOMCAT_STOP_SCRIPT || exit 0

start() {
    $ECHO -n "Starting Tomcat"
    su - $TOMCAT_USER -c "$TOMCAT_START_SCRIPT &"
    $ECHO "."
}

stop() {
    $ECHO -n "Stopping Tomcat"
    su - $TOMCAT_USER -c "$TOMCAT_STOP_SCRIPT 60 -force &"
    while [ "$(ps -fu $TOMCAT_USER | grep java | grep tomcat | wc -l)" -gt "0" ]; do
        sleep 5; $ECHO -n "."
    done
    $ECHO "."
}

case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        stop
        sleep 30
        start
        ;;
endcase
```

```

*)
    $ECHO "Usage: tomcat {start|stop|restart}"
    exit 1
esac
exit 0

```



The script must be changed if you use other user to start the service or your \$TOMCAT_HOME is not /home/openkm/tomcat

Make it executable:

```
$ sudo chmod 755 /etc/init.d/tomcat
```

Update the run-levels:

```
$ sudo chkconfig tomcat --level 2345 on
```

Check the service

Start the service:

```
$ sudo service tomcat start
```

Stop the service:

```
$ sudo service tomcat stop
```

Installing thirdparty software

Software	Required	Description
OpenOffice or LibreOffice	Yes	<p>Check these packages are present:</p> <ul style="list-style-type: none"> • openoffice.org-ure-3.1.1-19.5.el5_5.6 • openoffice.org-headless-3.1.1-19.5.el5_5.6 • openoffice.org-pyuno-3.1.1-19.5.el5_5.6 <p>Execute this command line to check the installed packages:</p> <pre>\$ rpm -qa grep openoffice</pre> <div> <p>Take the names of the packages as an orientation.</p> <p>If headless package is not installed OpenKM will not be able to start soffice service.</p> </div>

Tesseract	No	Some useful links: <ul style="list-style-type: none"> • https://code.google.com/p/tesseract-ocr/wiki/Compiling • https://code.google.com/p/python-tesseract/wiki/HowToCompileForCentos • http://www.vicchiam.com/blog/?p=168
ClamAV	No	Some useful links: <ul style="list-style-type: none"> • http://www.clamav.net/doc/install.html#rhel
Imagemagick	Yes	\$ yum install ImageMagick
GhostScript	Yes	\$ yum install ghostscript
Htop	no	Download from http://pkgs.repoforge.org/htop/ \$ rpm -ivH htop-1.0.3-1.el6.rf.x86_64.rpm

Start application

Check OpenKM.cfg parameters

```
$ vim /home/openkm/tomcat/OpenKM.cfg
```

Your configuration should be something like this:

```
# OpenKM Hibernate configuration values
hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect
hibernate.hbm2ddl=create
```



Ensure **hibernate.dialect=org.hibernate.dialect.MySQL5Dialect** is enabled if your database is MySQL

First time application start:

```
$ service tomcat start
```

To check if the startup process can execute the command:

```
$ tail -f /home/openkm/tomcat/log/catalina.log
```

When the application is started you'll see in log file the lines:

```
2015-07-04 18:28:10,680 [main] INFO org.apache.coyote.http11.Http11Protocol - Starti
2015-07-04 18:28:10,688 [main] INFO org.apache.coyote.ajp.AjpProtocol - Starting Pro
2015-07-04 18:28:10,692 [main] INFO org.apache.catalina.startup.Catalina - Server st
```

The application will be accessible at http://YOUR_IP:8080/OpenKM by user:**okmAdmin** with password:**admin**. (Do not forget to change it !). For more information please check: [Manage users and roles](#).

Configure the default extensions:

Go to Administration > [Database query](#) and execute:

```
INSERT INTO OKM_EXTENSION (EXT_UUID, EXT_NAME) VALUES ('808e7a42-2e73-470c-ba23-e4c9d
INSERT INTO OKM_EXTENSION (EXT_UUID, EXT_NAME) VALUES ('58392af6-2131-413b-b188-1851a
INSERT INTO OKM_PROFILE_MSC_EXTENSION (PEX_ID, PEX_EXTENSION) VALUES (1, '808e7a42-2e
INSERT INTO OKM_PROFILE_MSC_EXTENSION (PEX_ID, PEX_EXTENSION) VALUES (1, '58392af6-21
```

Linux Oracle 7.X Troubleshooting

The libreoffice and tesseract tools are not into the default repositories, you can follow the steps described in the URL below for installing them:

Install libreoffice from RPM



You should download the last RPM file version, what might be found at <https://www.libreoffice.org/download/download/?type=rpm-x86&version=6.0.4&lang=en>

The process description below has been described at <https://www.tecmint.com/install-libreoffice-on-rhel-centos-fedora-debian-ubuntu-linux-mint/>

Libreoffice at the end will be installed into the folder /opt/libreoffice6.0

```
$ yum remove openoffice* libreoffice*
$ cd /opt
$ wget https://www.libreoffice.org/donate/dl/rpm-x86_64/6.0.4/es/LibreOffice_6.0.4_Linux_x86-64_rpm.tar.gz
$ tar -xvf LibreOffice_6.0.4_Linux_x86-64_rpm.tar.gz
$ cd LibreOffice_6.0.4_Linux_x86-64_rpm/RPMS
$ yum localinstall *.rpm
```

Install Tesseract



The script below has been found from <https://github.com/EisenVault/install-tesseract-redhat-centos/> we suggest take a look at the last version of the script in the github project for updates.

Before executing the script, review it, consider to apply changes about downloading the dictionaries you need, for

example for spanish dictionary should be download <https://github.com/tesseract-ocr/tessdata/raw/3.04.00/spa.traineddata> what is not present in the script by default.

```
#!/bin/sh

cd /opt

yum -y update
yum -y install libstdc++ autoconf automake libtool autoconf-archive pkg-config gcc gcc-c++

#Install AutoConf-Archive
wget ftp://mirror.switch.ch/pool/4/mirror/epel/7/ppc64/a/autoconf-archive-2016.09.16-1.el7.noarch.rpm
rpm -i autoconf-archive-2016.09.16-1.el7.noarch.rpm

#Install Leptonica from Source
wget http://www.leptonica.com/source/leptonica-1.75.3.tar.gz
tar -zxvf leptonica-1.75.3.tar.gz
cd leptonica-1.75.3
./autobuild
./configure
make
make install
cd ..

#!/bin/sh

#Install Tesseract from Source
wget https://github.com/tesseract-ocr/tesseract/archive/3.05.01.tar.gz
tar -zxvf 3.05.01.tar.gz
cd tesseract-3.05.01
./autogen.sh
PKG_CONFIG_PATH=/usr/local/lib/pkgconfig LIBLEPT_HEADERSDIR=/usr/local/include ./configure
LDFLAGS="-L/usr/local/lib" CFLAGS="-I/usr/local/include" make
make install
ldconfig
cd ..

#Download and install tesseract language files
wget https://github.com/tesseract-ocr/tessdata/raw/3.04.00/ben.traineddata
wget https://github.com/tesseract-ocr/tessdata/raw/3.04.00/eng.traineddata
wget https://github.com/tesseract-ocr/tessdata/raw/3.04.00/hin.traineddata
wget https://github.com/tesseract-ocr/tessdata/raw/3.04.00/tha.traineddata
wget https://github.com/tesseract-ocr/tessdata/raw/3.04.00/osd.traineddata
mv *.traineddata /usr/local/share/tessdata

#Download Hindi Cube data
wget https://github.com/tesseract-ocr/tessdata/raw/3.04.00/hin.cube.bigrams
wget https://github.com/tesseract-ocr/tessdata/raw/3.04.00/hin.cube.fold
wget https://github.com/tesseract-ocr/tessdata/raw/3.04.00/hin.cube.lm
wget https://github.com/tesseract-ocr/tessdata/raw/3.04.00/hin.cube.nn
wget https://github.com/tesseract-ocr/tessdata/raw/3.04.00/hin.cube.params
wget https://github.com/tesseract-ocr/tessdata/raw/3.04.00/hin.cube.word-freq
wget https://github.com/tesseract-ocr/tessdata/raw/3.04.00/hin.tesseract_cube.nn
mv hin.* /usr/local/share/tessdata

ln -s /opt/tesseract-3.05.01 /opt/tesseract-latest
```

Other common additional setup configurations

OpenKM comes with a lot of configuration parameters we suggest taking a look at: [Recommended configuration parameters](#).

- [Configuring Apache HTTP Reverse-Proxy](#)

- [Configuring mail](#)
- [Configuring Tomcat memory utilization](#)
- [Configuring Tomcat port](#)
- [Configuring Tomcat SSL/TLS](#)
- [Configuring Tomcat timezone](#)

Restart installation from scratch

The OpenKM first startup does some background configurations like filesystem folder creation, changes in the OpenKM.cfg what only happens during the first startup.

Sometimes maybe some mistake in OpenKM configuration, for example, wrong database connection or similar issues during the first startup what raise and application failure. To restart the installation from scratch we suggest following the next steps.

1. Ensure OpenKM is stopped.
2. Edit the OpenKM.cfg and set the parameter **hibernate.hbm2ddl=create**
3. In case issue with database connection, we suggest checking the connection from the OpenKM server. Remember some databases require enabling the user to gain connection from a server (specific IP, range of IP, any server).
4. Delete the folder named "**repository**"
5. Start OpenKM.

Installing on Windows

Preliminaries

Check that you have enough disk size.

We recommend installing these tools:

- [Geany](#) (text editor).
- [mTail](#) (Windows Program that works like the Unix command "tail").

Checking Java version

```
C:\> java -version
```

If Java is correctly installed on your computer, the name and version of the Java virtual machine is displayed:

```
java version "1.8.0_101"  
Java(TM) SE Runtime Environment (build 1.8.0_101-b13)  
Java HotSpot(TM) 64-Bit Server VM (build 25.101-b13, mixed mode)
```



The Java version may be different depending on your installed one, but we always recommend the latest 1.8 release.

We recommend also setting the JAVA_HOME variable:

```
setx JAVA_HOME "C:\Program Files\Java\jdk1.8.0_101" /M
```

Install Java



Do these steps only if java 8 is not installed on your server.

If the required version of Java is not installed on your computer:

1. Download it from the Oracle website ([Java 8](#)) and choose the appropriate platform for your hardware and Windows version.
2. Run the downloaded .exe file and follow the instructions displayed.

Installing database

Install MySQL if you do not have it already installed:

Download MySQL MSI Installer from [MySQL Community Server](#) and follow the installation wizard steps.



If you do not have a MySQL client software it is also good idea to install [MySQL Workbench](#).

Check InnoDB is default MySQL engine

```
show engines;
```



It should show something like this:

| **InnoDB** | **DEFAULT** | Supports transactions, row-level locking, and foreign keys | YES | YES | YES |

If it not default, modify `c:\ProgramData\MySQL\MySQL Server 5.6\my.ini`

and under `[mysqld]` add

```
default-storage-engine = innodb
```



After changing default MySQL engine the MySQL service must be restarted to take effect.

Create database

```
CREATE DATABASE okmdb DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_bin;  
CREATE USER openkm@localhost IDENTIFIED BY 'password';  
GRANT ALL ON okmdb.* TO openkm@localhost WITH GRANT OPTION;
```

Installing OpenKM and Tomcat bundle

Unzip `openkm-6.3.2-community-tomcat-bundle.zip` to `c:\`



The Tomcat version numeration name can be altered in order to be upgraded to major version.

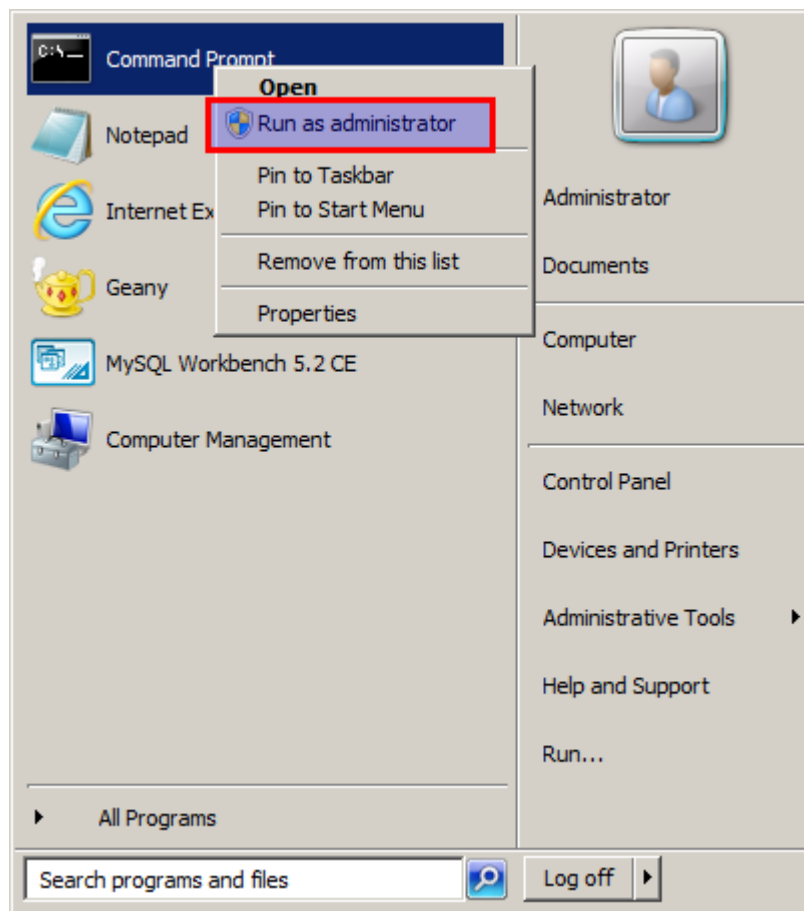
Copy files from `C:\tomcat\bin\x86` or `C:\tomcat\bin\x64` to `C:\tomcat\bin` based on the server architecture.

Configuring Tomcat as a service

Register the service:



Open a windows terminal as administrator.



```
cd c:\tomcat\bin
service.bat install
```

Set the JVM parameters for the Tomcat service:

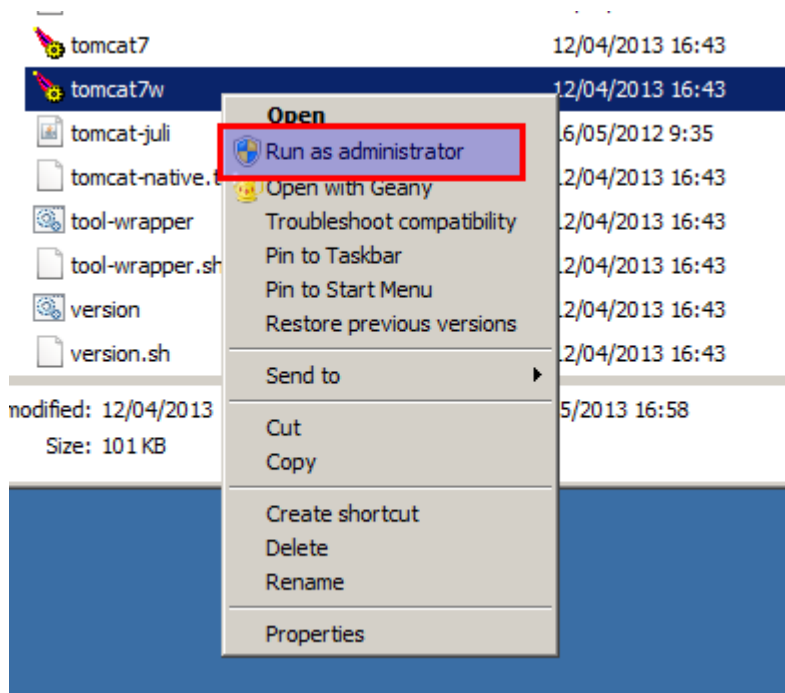


The parameters below are deprecated when you are running OpenJDK 1.8. do not execute these ones:

```
openkm //US//OpenKM ++JvmOptions="-XX:PermSize=256m"
openkm //US//OpenKM ++JvmOptions="-XX:MaxPermSize=512m"
```

```
cd c:\tomcat-7.0.61\bin
openkm //US//OpenKM ++JvmOptions="-XX:+UseG1GC"
openkm //US//OpenKM ++JvmOptions="-Djava.awt.headless=true"
openkm //US//OpenKM ++JvmOptions="-Djava.net.preferIPv4Stack=true"
openkm //US//OpenKM ++JvmOptions="-Dfile.encoding=utf-8"
openkm //US//OpenKM ++JvmOptions="-Dsun.jnu.encoding=UTF-8"
openkm //US//OpenKM ++JvmOptions="-Dmail.mime.ignoreunknownencoding=true"
```

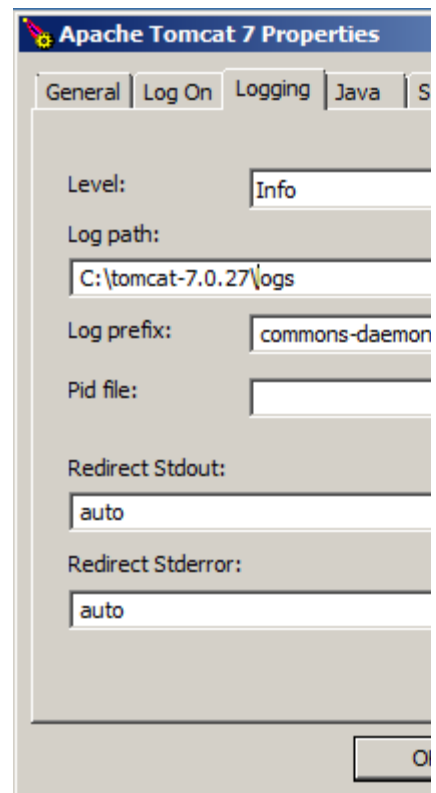
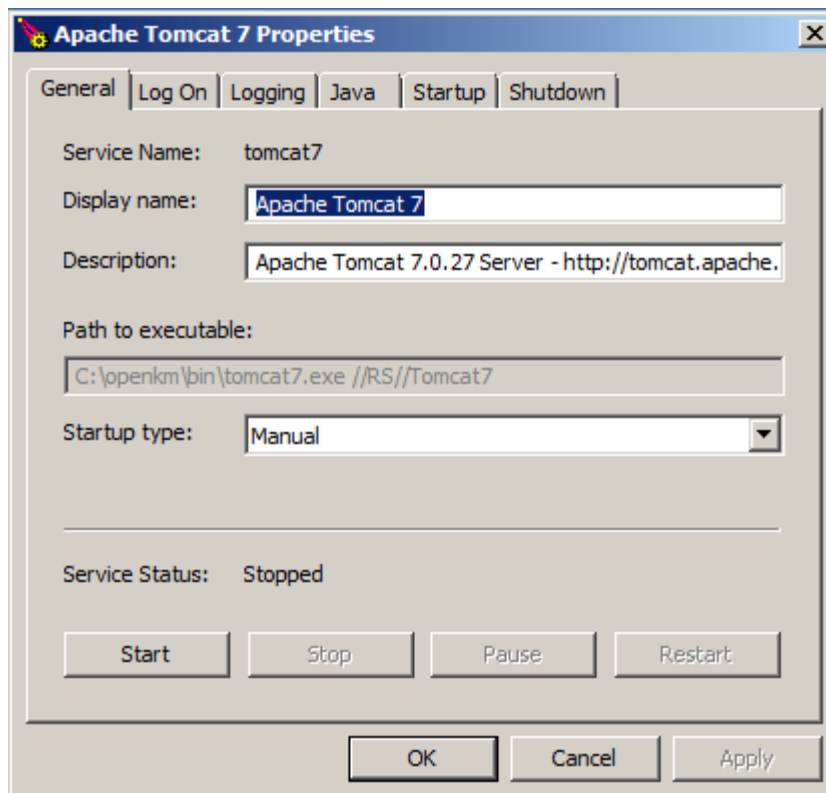
Execute the tomcat7w.exe as administrator:

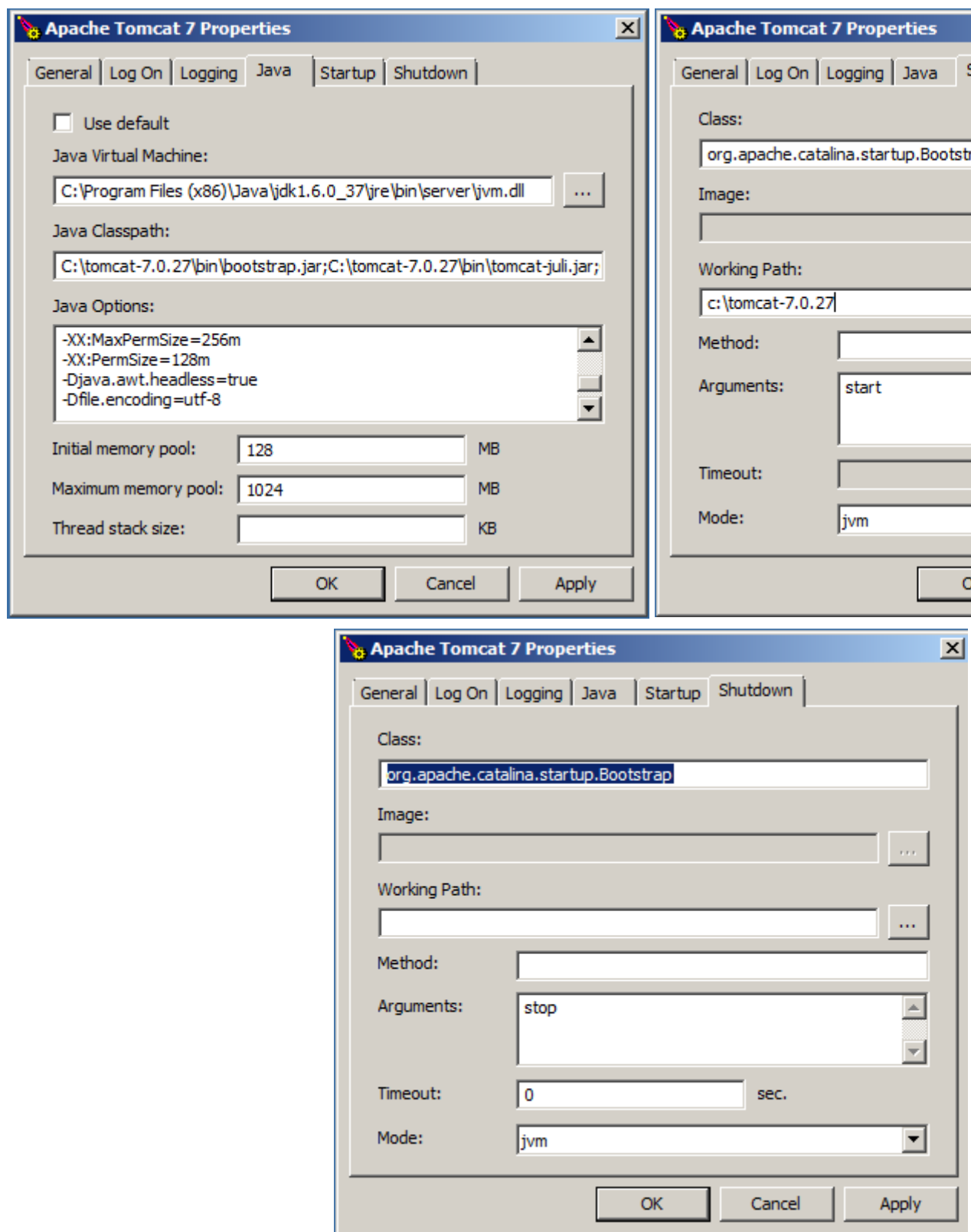


Check the configuration parameters:



Also you can change "Startup type" to Automatic.





More information about how to configure Tomcat service on Windows at: <http://tomcat.apache.org/tomcat-7.0-doc/windows-service-howto.html>

Check the service

From terminal

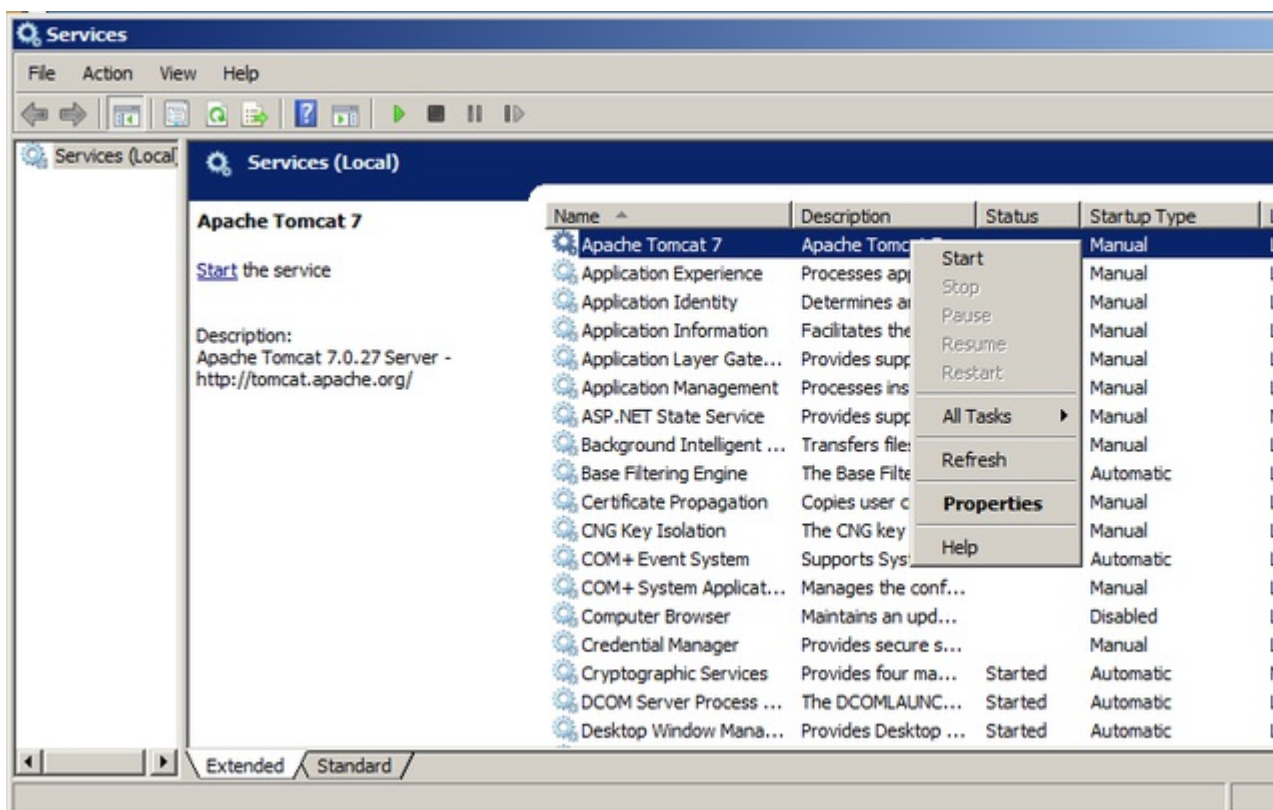
Start the service:

```
$ net start tomcat7
```

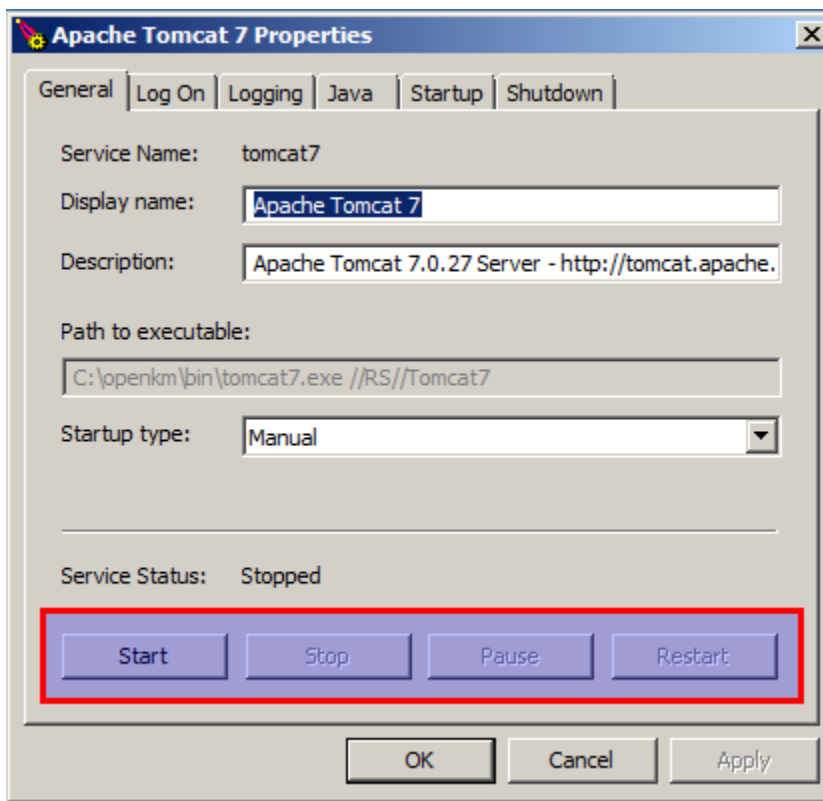
Stop the service:

```
$ net stop tomcat7
```

From Windows service panel



From tomcat7w tool



Configuring environment variables

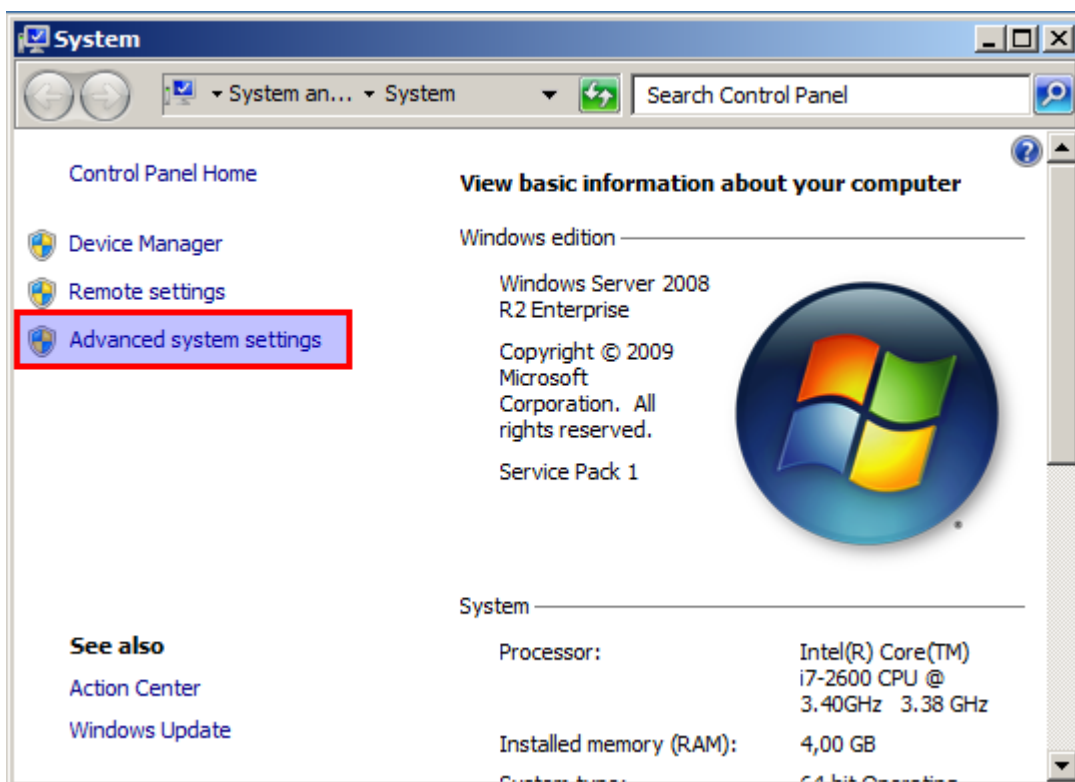
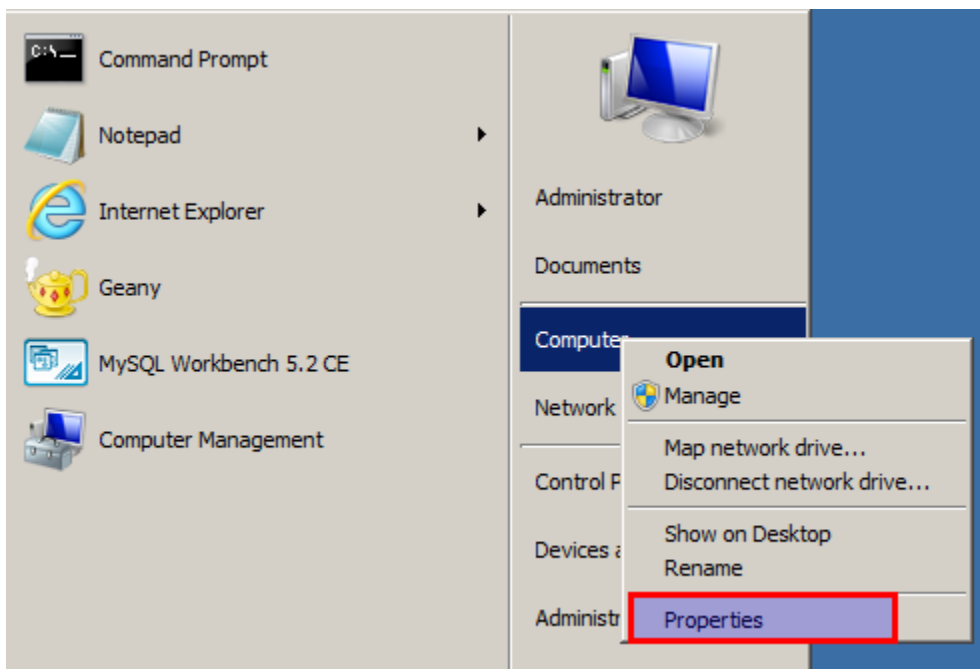
Add to environment path variable the **sigar** folder, the **bin** folder:

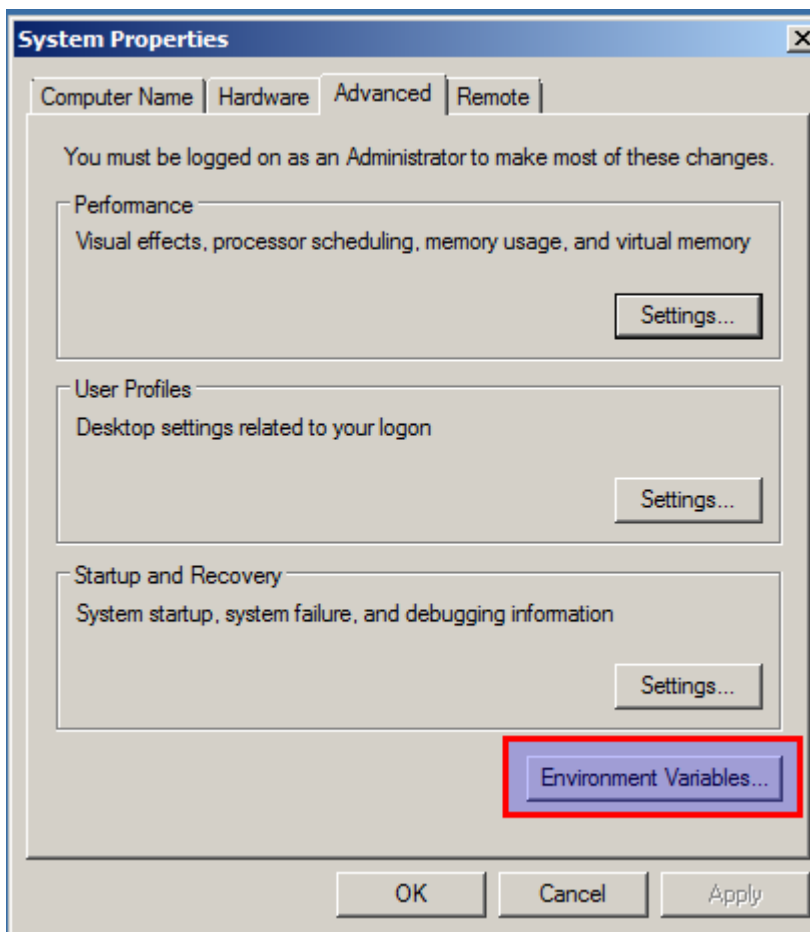
```
setx PATH "%PATH%;C:\tomcat\lib\sigar" /M
setx PATH "%PATH%;C:\tomcat\bin" /M
```

These operations also can be done from Windows environment variables:

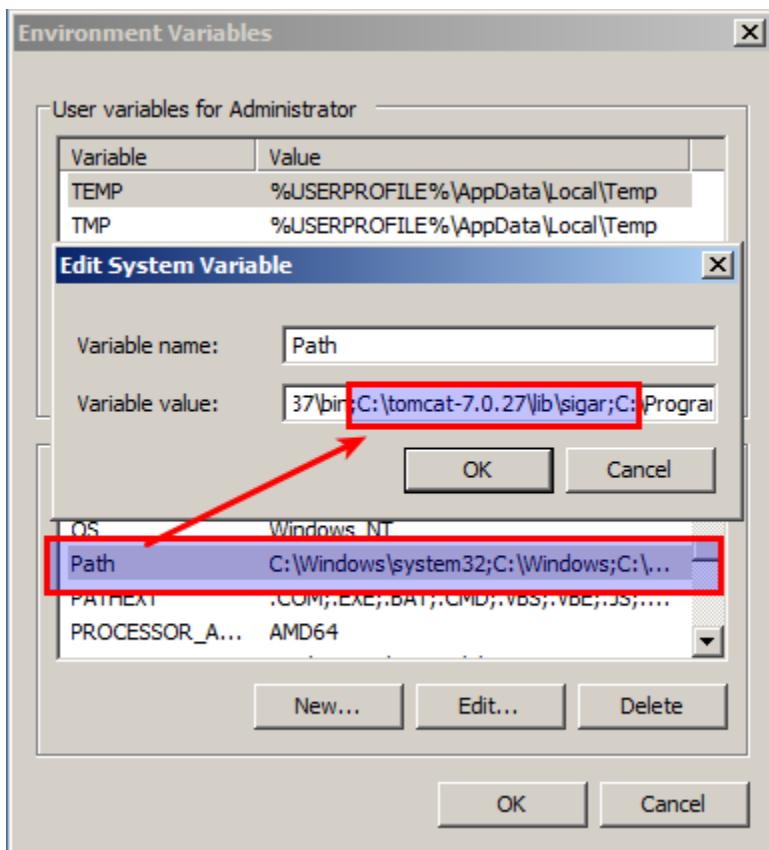


Depending on your Windows OS version the next screens could be quite different.





Add sigar folder path - `c:\tomcat\lib\sigar\` - to your environment path variable:



Installing OpenKM

With Tomcat stopped copy the OpenKM.war file into /home/openkm/tomcat/webapps

```
$ copy OpenKM.war c:\tomcat-7.0.61\webapps
```

Install third party software:

Software	Required	Description
OpenOffice	Yes	We recommend using OpenOffice rather than Libreoffice, because we've found problems in some servers to setup as service by OpenKM.
Tesseract	No	Download from https://code.google.com/p/tesseract-ocr/ and follow the installation wizard.
Clamav	No	Download from http://www.clamav.net/download.html and follow the installation wizard.
Imagemagick	No	Tomcat bundle comes with embedded convert tool.

		<p>i In case you decide for install it, go to http://www.imagemagick.org/script/binary-releases.php and download the latest binary release and follow the installation wizard.</p> <p>Take in consideration 16 bits needs more hardware resources than 8 bits.</p> <p>We suggest ImageMagick-6.9.1-7-Q8-x64-dll.exe rather than ImageMagick-6.9.1-7-Q16-x64-dll.exe (note the version numeration can be distinct due ImageMagick version upgrade).</p>
GhostScript	No	<p>Tomcat bundle comes with embedded Ghostscript tool.</p> <p>If you decide installing it, go to http://ghostscript.com/download/gsdnld.html and follow the installation wizard.</p>
There is also an available Extras.zip file that contains all these software as a Portable edition.		

Start application

Check OpenKM.cfg parameters

```
C:\> geany c:\openkm\OpenKM.cfg
```

Your configuration should be something like this:

```
# OpenKM Hibernate configuration values

hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect

hibernate.hbm2ddl=create

# Initial configuration - Windows

system.openoffice.path=C:/Program Files (x86)/OpenOffice 4
```



Ensure **hibernate.dialect=org.hibernate.dialect.MySQL5Dialect** is enabled if your database is MySQL.

First time application start:

```
$ net start tomcat7
```



Use mTail Program - or equivalent - to watch the c:\tomcat-7.0.61\log\catalina.log file

When application is started you'll see it in the log file the lines:

```
2015-07-04 18:28:10,680 [main] INFO org.apache.coyote.http11.Http11Protocol - Starti
2015-07-04 18:28:10,688 [main] INFO org.apache.coyote.ajp.AjpProtocol - Starting Pro
2015-07-04 18:28:10,692 [main] INFO org.apache.catalina.startup.Catalina - Server st
```

The application will be accessible at http://YOUR_IP:8080/OpenKM by user:**okmAdmin** with password:**admin**. (Do not forget to change it !). For more information: [Manage users and roles](#).

Configure default extensions:

Go to Administration > [Database query](#) and execute:

```
INSERT INTO OKM_EXTENSION (EXT_UUID, EXT_NAME) VALUES ('808e7a42-2e73-470c-ba23-e4c9d
INSERT INTO OKM_EXTENSION (EXT_UUID, EXT_NAME) VALUES ('58392af6-2131-413b-b188-1851a
INSERT INTO OKM_PROFILE_MSC_EXTENSION (PEX_ID, PEX_EXTENSION) VALUES (1, '808e7a42-2e
INSERT INTO OKM_PROFILE_MSC_EXTENSION (PEX_ID, PEX_EXTENSION) VALUES (1, '58392af6-21
```

Other common additional setup configurations

OpenKM comes with a lot of configuration parameters please take a look at: [Recommended configuration parameters](#).

- [Configuring Apache HTTP Reverse-Proxy](#)
- [Configuring mail](#)
- [Configuring Tomcat memory utilization](#)
- [Configuring Tomcat port](#)
- [Configuring Tomcat SSL/TLS](#)
- [Configuring Tomcat timezone](#)

Restart installation from scratch

The OpenKM first startup does some background configurations like filesystem folder creation, changes in the OpenKM.cfg what only happens during the first startup.

Sometimes maybe some mistake in OpenKM configuration, for example, wrong database connection or similar issues during the first startup what raise and application failure. To restart the installation from scratch we suggest following the next steps.

1. Ensure OpenKM is stopped.
2. Edit the OpenKM.cfg and set the parameter **hibernate.hbm2ddl=create**
3. In case issue with database connection, we suggest checking the connection from the OpenKM server. Remember some databases require enabling the user to gain connection from a server (specific IP, range of IP, any server).
4. Delete the folder named "**repository**"
5. Start OpenKM.

Using the installer

Another way of installing OpenKM is using the installer tool. This is a Java application which eases the installation process and works on Ubuntu, Debian, RedHat, CentOS and Windows platforms. Keep in mind that you need a working Java installation prior to use this tool. We recommend OpenJDK 8. Please check these pages for more information about installing Java on your server; look for **Checking Java version** section:

- [Installing on RedHat and CentOS](#)
- [Installing on Ubuntu and Debian](#)
- [Installing on Windows](#)



The installer downloads several packages from SourceForge like Tomcat and OpenKM, so it should have connection access.

Once Java 8 is installed you can download the installer from

<https://sourceforge.net/projects/openkm/files/common/OKMInstaller.jar>. In Linux you can download it from the command line:

```
$ wget https://sourceforge.net/projects/openkm/files/common/OKMInstaller.jar
```



The installer will make the installation in the folder which it runs in, so before running move the OKMInstaller.jar to the location where OpenKM should be installed. In case of running on Linux it also changes the files permissions to a user called "openkm" so, if you haven't created such user, please create it now:

```
$ sudo adduser openkm
```

This command should work in most common Linux distributions.

It's a text-based installer, so you need to execute it from console as root if you are on Linux or as Administrator if you are on Windows. This is because it needs to install package dependencies and configure OpenKM as a service:

```
$ java -jar OKMInstaller.jar
```

And this is the output:

```
Running in Linux: 0.2.3 (build: 25467)
Linux distro: ubuntu (Ubuntu - 14.04)
Java version: 1.8.0_77
Current directory: /home/openkm

Database [mariadb]:
Database root password [wmgos1ioQY]:
Database name [okmdb]:
Database user [openkm]:
Database password [F9pqDM6kTk]:
-----
- Database: mariadb
- Install database: true
- Database root password: wmgos1ioQY
- Database name: okmdb
- Database user: openkm
- Database password: F9pqDM6kTk
- Install version: 6.4.27
-----
Start install process? [y/N]:
```

Values in brackets are the default one. If you hit Enter, this is the chosen value. If you want a different one, just write the string. At the end, the installed will show the entered information and you would be able to verify it and proceed or enter again these values.

Once the info is validated, the installation process will start. This process may take several minutes depending on your Internet connection and server performance. In case of Linux based systems, it will install all required dependencies like ImageMagick and LibreOffice.



If you access the Internet using a proxy, just configure it when executing the application:

```
$ java -Dhttp.proxyHost=192.168.0.1 -Dhttp.proxyPort=3128 -jar OKMInstaller.jar
```

You may also need **https.proxyHost** and **https.proxyPort** or even **http.proxyUser**, **http.proxyPassword**, **https.proxyUser** and **https.proxyPassword** parameters.

Supported databases

The installer supports these databases:

- [MariaDB](#) (mariadb)
- [MySQL](#) (mysql)
- [PostgreSQL](#) (postgres)
- [SQL Server](#) (sqlserver)
- [Oracle](#) (oracle)
- [HSQLDB](#) (hsqldb)

- [H2](#) (h2)

Other options

The installer will also check if the port 8080 is already being used, which is the default port used by Tomcat and should be free.

If you want to disable this check, use the **-p** or **--port** parameters.

```
$ java -jar OKMInstaller.jar -p
```


Using Docker

Docker is a software platform that allows you to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime. Using Docker, you can quickly deploy and scale applications into any environment and know your code will run.

There is a public Docker image available at <https://hub.docker.com/repository/docker/openkm/openkm-ce> and it is the easiest way to get OpenKM running on your server.

Please, refer to [Docker documentation](#) for installation instructions and to learn more about Docker.

Once Docker has been installed, just execute this command to get your OpenKM CE instance:

```
$ docker run --name openkm-ce -p 8080:8080 openkm/openkm-ce
```

And after that, point your browser at <http://localhost:8080/>. The default user and password is "okmAdmin" / "admin".

Database

OpenKM stores most of its data in a SQL database. Several databases are supported, but they must be configured in order to work correctly.

Configure database

- [Configuring MySQL and MariaDB.](#)
- [Configuring Oracle.](#)
- [Configuring PostgreSQL.](#)
- [Configuring SQL Server.](#)
- [Configuring other databases.](#)

Configuring MySQL and MariaDB



MariaDB is a community-developed split of the MySQL intended to remain free under the GNU GPL. Being a split of a leading Open Source software system, it is notable for being led by the original developers of MySQL, who separated it due to concerns over its acquisition by Oracle.

Preliminaries



Perform all changes with the application stopped.

Check if your MySQL installation has InnoDB engine enabled:

```
$ mysql -h localhost -u root -p
> show engines;
```



It should show something like this:

| **InnoDB** | **DEFAULT** | Supports transactions, row-level locking, and foreign keys | YES | YES | YES |

In case of another default engine, there are two options:

1. Use the **org.hibernate.dialect.MySQL5InnoDBDialect** dialect and avoid changing the default MySQL Storage Engine.
2. Change default engine

Change default engine

Modify the MySQL configuration file named **my.cnf**

and under [mysqld] add

```
default-storage-engine = innodb
```



After changing the default to MySQL the service must be restarted to take effect.

Change default charset

In Ubuntu 16.04 the default charset configured in MariaDB and MySQL is **utf8mb4** but we recommend switching to **utf8**. In order to get the recommended charset you have to modify these files:

/etc/mysql/mariadb.conf.d/50-server.cnf

```
character-set-server = utf8
```

```
collation-server = utf8_general_ci
```

/etc/mysql/mariadb.conf.d/50-client.cnf

```
default-character-set = utf8
```

/etc/mysql/mariadb.conf.d/50-mysql-clients.cnf

```
default-character-set = utf8
```

Once modified, a database server restart is needed.

Database creation

Create a database and user

```
DROP DATABASE IF EXISTS okmdb;  
CREATE DATABASE okmdb DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_bin;  
CREATE USER openkm@localhost IDENTIFIED BY '*secret*';  
GRANT ALL ON okmdb.* TO openkm@localhost WITH GRANT OPTION;
```

You can check the database engine with:

```
$ mysqlshow -h localhost -u root -p --status okmdb;
```

More information at [MySQL: Case Sensitivity in String Searches](#)

Configure your OpenKM.cfg



This configuration property should be set before the database creation. Once the database has been initialized don't modify it because can damage your installation.

If your OpenKM installation has been already configured with another database you can't switch to another database by simply changing this property.

Edit the file \$TOMCAT_HOME/OpenKM.cfg

```
hibernate.dialect=org.hibernate.dialect.MySQL5Dialect  
hibernate.hbm2ddl=create
```

or (recommended)

```
hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect  
hibernate.hbm2ddl=create
```

MySQL timezone

If you want to set your timezone in your jdbc connection according to your timezone configured in your system, first you can

see the mappings both for MySQL and for MariaDB in this way:

```
SELECT * FROM mysql.time_zone_name
```

So, for example, you could configure your jdbc connection like this if your system timezone is for example UTC+1:

```
spring.datasource.url=jdbc:mysql://localhost:3306/okmdb?
autoReconnect=true&useUnicode=true&characterEncoding=UTF8&serverTimezone=Europe/Madrid
```

Where **Europe/Madrid** is a valid value taken from the table **time_zone_name**.

Configure Tomcat datasources

Edit the file \$TOMCAT_HOME/conf/server.xml and enable the resource named jdbc/OpenKMDS

MySQL:

```
<Resource name="jdbc/OpenKMDS" auth="Container" type="javax.sql.DataSource"
maxActive="100" maxIdle="30" maxWait="10000" validationQuery="select 1"
username="openkm" password="*secret*" driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost:3306/okmdb?autoReconnect=true&useUnicode=true&characterEncoding=UTF8&serverTimezone=Europe/Madrid"/>
```

MariaDB:

```
<Resource name="jdbc/OpenKMDS" auth="Container" type="javax.sql.DataSource"
maxActive="100" maxIdle="30" maxWait="10000" validationQuery="select 1"
username="openkm" password="*secret*" driverClassName="org.mariadb.jdbc.Driver"
url="jdbc:mariadb://localhost:3306/okmdb?useSSL=false&autoReconnect=true&characterEncoding=UTF8&serverTimezone=Europe/Madrid"/>
```



MariaDB can use these driverClassName values:

- com.mysql.jdbc.Driver
- org.mariadb.jdbc.Driver

In recent MySQL versions, it seems you need to add **useSSL=false** to the connection URL.



Configures the attributes values named:

- userName.
- password.
- url (change to your hosts and port).

It can only use a resource named jdbc/OpenKMDS.

Configure application login

Edit the file \$TOMCAT_HOME/OpenKM.xml

```
<security:authentication-manager alias="authenticationManager">
  <security:authentication-provider>
    <security:password-encoder hash="md5"/>
    <security:jdbc-user-service
      data-source-ref="dataSource"
      users-by-username-query="select usr_id, usr_password, 1 from OKM_USER where u.
      authorities-by-username-query="select ur_user, ur_role from OKM_USER_ROLE whe
    </security:authentication-provider>
  </security:authentication-manager>
```

Check for JDBC Driver

Go to \$TOMCAT_HOME/lib and check for JDBC **mysql-connector-java-5.1.12-bin.jar** Driver.

If it is not present, download MySQL JDBC driver from [MySQL Home Page](#) and move it.



The driver numeration version could be distinct due some version upgrade.



MariaDB can use its own JDBC mariadb-java-client-1.1.7.jar Driver.

If is not present, download mariaDB JDBC driver from [MariaDB Connector/J](#) and move it.

Run application

As a process into OpenKM startup, the application will create automatically and empty database structure.



Once the tables are created, OpenKM will automatically change the **hibernate.hbm2ddl** property from *create* to *none*.

Additional information

- [InnoDB Performance Tuning Tips](#)
- [InnoDB - High disk write I/O on ibdata1 file and ib_logfile0](#)

Remove and create MySQL service in Windows



Ensure your mysql/bin folder is in your PATH at environment variables.

For example in MySQL 5.7 the bin folder location is at C:\Program Files\MySQL\MySQL Server 5.7\bin

If you have MySQL Window service you can remove with one of these command line:

```
c:\> mysql --remove MySQL57
```

or

```
c:\> sc delete MySQL57
```

Create the service:

```
c:\> "C:\Program Files\MySQL\MySQL Server 5.7\bin\mysqld.exe" --install MYSQL57 --defaults-  
file="C:\ProgramData\MySQL\MySQL Server 5.7\my.ini"
```



More information at <http://dev.mysql.com/doc/refman/5.7/en/windows-start-service.html>

Migrate from Windows to Linux



By default table names in Windows are lowercase and in Linux in uppercase.

When you migrate from Windows to Linux you must convert lowercase table name to uppercase. The next script will help on it:

```
select concat('rename table ', table_name, ' to ' , upper(table_name) , ';') from inf
```

MySQL tuning

Take this documentation as is, not as a definitive guide to tuning MySQL. There is a lot of information and knowledge that should be read before changing MySQL configuration. The documentation has been created based on experience shared by OpenKM server administrator users.

Useful links should read:

- [MySQL: InnoDB Performance Tuning Tips](#)
- [MySQL: The MySQL Query Cache](#)
- [InnoDB - High disk write I/O on ibdata1 file and ib_logfile0](#)

Sample one

Server information

- Virtualized Windows Server 2008
- CPU: Intel Xeon X3470 @2,93GHz
- RAM: 7GB
- JVM settings for Tomcat: -Xms 3072 -Xmx 3072 -XX:PermSize=256m -XX:MaxPermSize=512m
- Repository is 1GB

MySQL my.cnf changes

- `innodb_buffer_pool_size=1792M`
- `innodb_additional_mem_pool_size=16M`
- `innodb_log_file_size=256M`
- `innodb_log_buffer_size=8M`

Using MySQL server InnoDB engine with the default configuration parameters in my.cnf file can cause a serious performance bottleneck in applications with high database read write activity. MySQL should have enough RAM to be able to cache the repository and avoid the high Disk I/O activity.

The most important parameters are:

- `innodb_buffer_pool_size`
- `innodb_additional_mem_pool_size`
- `innodb_log_file_size`
- `innodb_log_buffer_size`



According to some database experts, `innodb_buffer_pool_size` should occupy the 70 percent of RAM of a dedicated database server.



Before changing these values, please read MySQL documentation carefully. Before making any changes you should make sure that MySQL server has been stopped correctly (check server log), otherwise the innodb will become corrupted and MySQL server won't restart.

Considerations Repository is 1GB so 1792M is enough RAM for `innodb_buffer_pool_size`. The larger the repository the more memory you need to allocate to `innodb_buffer_pool_size`. If you are using x86 MySQL server you cannot allocate RAM larger than 2GB.

Information collected from [OpenKM forum post](#).

Sample two

Server information

- CPU: Intel Xeon E5606 @2.13GHz
- RAM: 4GB
- System: Windows server 2008, 64bit
- JAVA_OPTS=-Xms256m -Xmx1024m -XX:PermSize=256m -XX:MaxPermSize=512m -Djava.awt.headless=true -Dfile.encoding=utf-8
- Database size:2.5G
- Repository size:40GB
- 100,000 nodes(folders + documents)

MySQL my.cnf changes

- `query_cache_size=0`
- `table_open_cache=2000`
- `tmp_table_size=16M`
- `thread_cache_size=9`
- `myisam_max_sort_file_size=100G`
- `myisam_sort_buffer_size=32M`
- `key_buffer_size=8M`
- `read_buffer_size=64K`
- `read_rnd_buffer_size=256K`
- `sort_buffer_size=256K`
- `innodb_additional_mem_pool_size=16M`
- `innodb_log_buffer_size=8M`

- `innodb_buffer_pool_size=1042M`

Results

Before optimization the login time was about 45-55 seconds with a lot of hard-disk I/O usage. After the MySQL optimization login takes about 10 seconds.



With more RAM it will be possible to get a higher MySQL buffer and probably better performance.

Information collected [OpenKM Forum post](#).

Change MySQL data directory

Ubuntu

Stop the service:

```
$ service mysql stop
```

Create a new folder destination:

```
$ mkdir /new_path/mysql_data  
$ cp -R -p /var/lib/mysql /new_path/mysql_data
```



MySQL user should have enough credentials granted to the new folder /new_path/mysql_data.

Optionally modify the path of older mysql data store folder location:

```
$ mv /var/lib/mysql /var/lib/mysql_old
```

Edit the **datadir** property into the **/etc/mysql/my.cnf** configuration file:

```
$ vim /etc/mysql/my.cnf
```

Change the older data store folder path with the newer:

```
datadir = /var/lib/mysql
```

to

```
datadir = /new_path/mysql_data
```

Enable apparmor. Edit the file **/etc/apparmor.d/usr.sbin.mysqld**:

```
$ vim /etc/apparmor.d/usr.sbin.mysqld
```

Change the older data store folder path with the newer:

```
/usr/lib/mysql/plugin/ r,  
/usr/lib/mysql/plugin/*.so* mr,  
...  
/var/lib/mysql/ r,  
/var/lib/mysql/** rwk,
```

to

```
/new_path/mysql_data/plugin/ r,  
/new_path/mysql_data/plugin/*.so* mr,  
...  
/new_path/mysql_data/ r,  
/new_path/mysql_data/** rwk,
```

Start MySQL service:

```
$ service mysql start
```



More information at <http://www.ubuntugeek.com/how-to-change-the-mysql-data-default-directory.html>.

CentOS and RHEL

Stop the service:

```
$ service mysqld stop
```

Create a new folder destination:

```
$ mkdir /new_path/mysql_data  
$ cp -R -p /var/lib/mysql /new_path/mysql_data
```



The mysql user must have enough credentials granted to the new folder /new_path/mysql_data.

Optionally modify the path of older mysql data store folder location:

```
$ mv /var/lib/mysql /var/lib/mysql_old
```

Edit the **datadir** property into the **/etc/mysql/my.cnf** configuration file:

```
$ vim /etc/mysql/my.cnf
```

Change the older data store folder path with the newer:

```
datadir = /var/lib/mysql
```

to

```
datadir = /new_path/mysql_data
```

Enable apparmor.

```
$ yum install policycoreutils-python
```

```
$ getenforce
$ semanage fcontext -a -t mysqld_db_t "/new_path/mysql_data(/.*)?"
$ restorecon -Rv /new_path/mysql_data
```

Start MySQL service:

```
$ service mysqld start
```

More information at <http://lifeofageekadmin.com/how-to-change-the-default-mysql-data-directory-on-rhel-6/>

Windows

Stop the service:

```
C:\> net stop mysql56
```

Create a new folder destination:

```
C:\> robocopy "C:\ProgramData\MySQL\MySQL Server 5.6\data" c:\new_path\mysql_data
```

Optionally modify the path of older mysql data store folder location:

```
C:\> ren "C:\ProgramData\MySQL\MySQL Server 5.6\data" "C:\ProgramData\MySQL\MySQL Server 5.6\data_old"
```

Edit the **datadir** property into the **C:\ProgramData\MySQL\MySQL Server 5.6\my.ini** configuration file:

Change the older data store folder path with the newer:

```
datadir = C:\ProgramData\MySQL\MySQL Server 5.6\data
```

to

```
datadir = c:\new_path\mysql_data
```

Start MySQL service:

```
C:\> net start mysql56
```

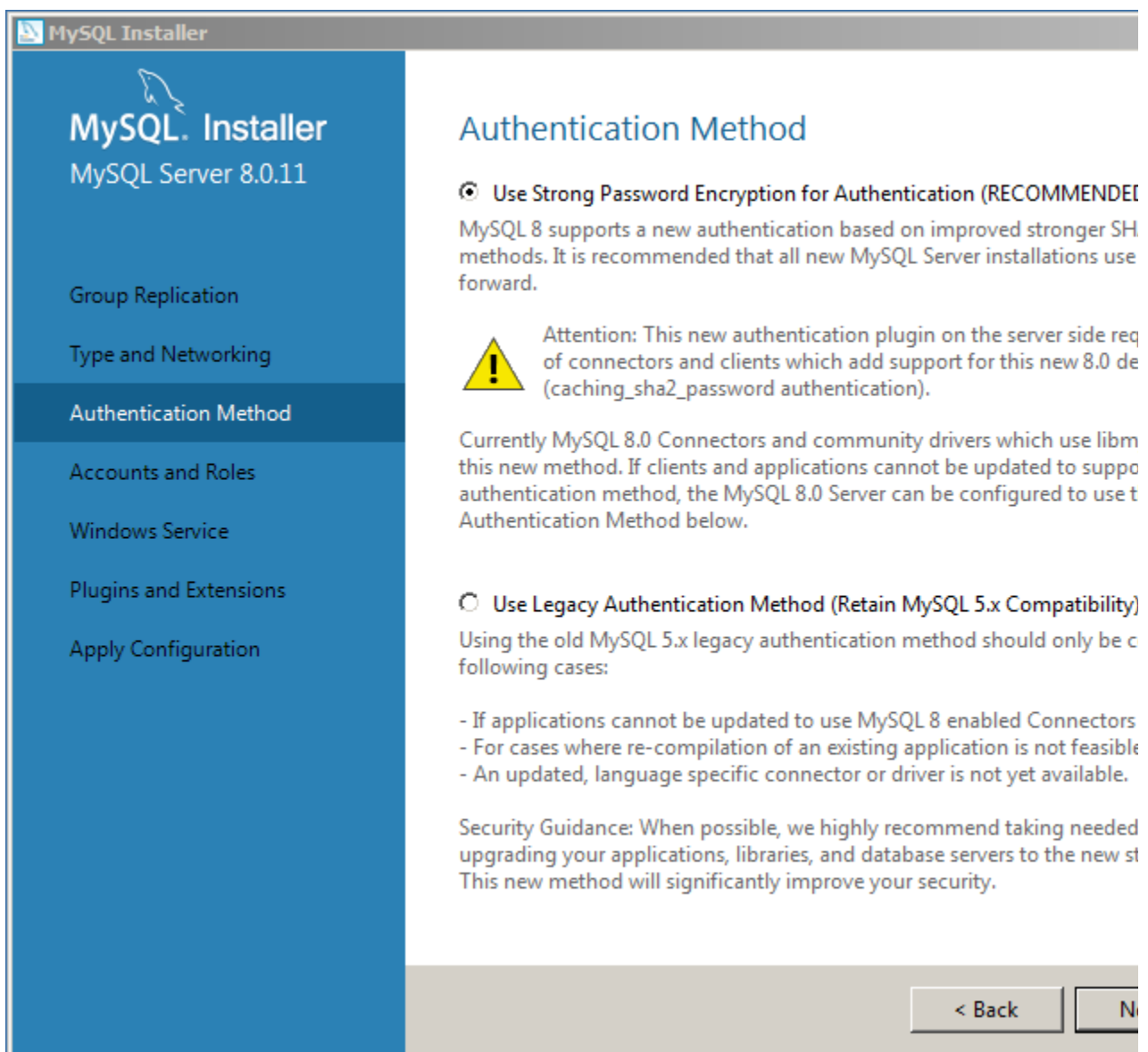
MySQL 8 troubleshooting

Could not create connection to database server

Might be several reason why the application is not able to stablish the connection to the database, take a look into the log file what might give to you the clues about what is happening.

Strong Password Encryption for Authentication

The latest MySQL installation wizard suggest you to use "**Strong Password Encryption for Authentication**" as can be shown in the next screenshot:



When the option "**Strong Password Encryption for Authentication**" is enabled rather than "**Legacy Authentication Method**"

you are enabling **SSL connection by default**. Than means you might read some literature and apply some changes in your

connection setting into your server.xml server to connect with the database. Usually in this scenario you will get some error like what is shown below:

```
java.sql.SQLException: Cannot create PoolableConnectionFactory (Could not create conn
at org.apache.tomcat.dbcp.dbcp2.BasicDataSource.createPoolableConnectionFactory(Basi
at org.apache.tomcat.dbcp.dbcp2.BasicDataSource.createDataSource(BasicDataSource.jav
at org.apache.tomcat.dbcp.dbcp2.BasicDataSource.getConnection(BasicDataSource.java:1
at org.hibernate.connection.DataSourceConnectionProvider.getConnection(DataSourceCon
at org.hibernate.cfg.SettingsFactory.buildSettings(SettingsFactory.java:113) ~[hiber
at org.hibernate.cfg.Configuration.buildSettingsInternal(Configuration.java:2863) [h
at org.hibernate.cfg.Configuration.buildSettings(Configuration.java:2859) [hibernate
at org.hibernate.cfg.Configuration.buildSessionFactory(Configuration.java:1870) [hib
at com.openkm.dao.HibernateUtil.getSessionFactory(HibernateUtil.java:367) [classes/
at com.openkm.dao.HibernateUtil.getSessionFactory(HibernateUtil.java:182) [classes/
at com.openkm.servlet.RepositoryStartupServlet.init(RepositoryStartupServlet.java:99
at javax.servlet.GenericServlet.init(GenericServlet.java:158) [servlet-api.jar:3.1.F
at org.apache.catalina.core.StandardWrapper.initServlet(StandardWrapper.java:1183) [
at org.apache.catalina.core.StandardWrapper.loadServlet(StandardWrapper.java:1099) [
at org.apache.catalina.core.StandardWrapper.load(StandardWrapper.java:989) [catalina
at org.apache.catalina.core.StandardContext.loadOnStartup(StandardContext.java:4931)
at org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:5241)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:150) [catalina.ja
at org.apache.catalina.core.ContainerBase.addChildInternal(ContainerBase.java:752) [
at org.apache.catalina.core.ContainerBase.addChild(ContainerBase.java:728) [catalina
at org.apache.catalina.core.StandardHost.addChild(StandardHost.java:734) [catalina.j
at org.apache.catalina.startup.HostConfig.deployWAR(HostConfig.java:952) [catalina.j
at org.apache.catalina.startup.HostConfig$DeployWar.run(HostConfig.java:1823) [catal
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) [na:1.8.0
at java.util.concurrent.FutureTask.run(FutureTask.java:266) [na:1.8.0_171]
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [
at java.lang.Thread.run(Thread.java:748) [na:1.8.0_171]
Caused by: java.sql.SQLNonTransientConnectionException: Could not create connection t
at com.mysql.cj.jdbc.exceptions.SQLException.createSQLException(SQLException.java:108) ~[mys
at com.mysql.cj.jdbc.exceptions.SQLException.createSQLException(SQLException.java:95) ~[mysq
at com.mysql.cj.jdbc.exceptions.SQLException.createSQLException(SQLException.java:87) ~[mysq
at com.mysql.cj.jdbc.exceptions.SQLException.createSQLException(SQLException.java:61) ~[mysq
at com.mysql.cj.jdbc.exceptions.SQLException.createSQLException(SQLException.java:71) ~[mysq
at com.mysql.cj.jdbc.ConnectionImpl.connectWithRetries(ConnectionImpl.java:932) ~[my
at com.mysql.cj.jdbc.ConnectionImpl.createNewIO(ConnectionImpl.java:857) ~[mysql-con
at com.mysql.cj.jdbc.ConnectionImpl.<init>(ConnectionImpl.java:444) ~[mysql-connecto
at com.mysql.cj.jdbc.ConnectionImpl.getInstance(ConnectionImpl.java:230) ~[mysql-con
at com.mysql.cj.jdbc.NonRegisteringDriver.connect(NonRegisteringDriver.java:226) ~[m
at org.apache.tomcat.dbcp.dbcp2.DriverConnectionFactory.createConnection(DriverConne
at org.apache.tomcat.dbcp.dbcp2.PoolableConnectionFactory.makeObject(PoolableConnect
at org.apache.tomcat.dbcp.dbcp2.BasicDataSource.validateConnectionFactory(BasicDataS
at org.apache.tomcat.dbcp.dbcp2.BasicDataSource.createPoolableConnectionFactory(Basi
... 27 common frames omitted
Caused by: com.mysql.cj.exceptions.CJCommunicationsException: Communications link fai
```

```
The last packet sent successfully to the server was 0 milliseconds ago. The driver ha
at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method) ~[na:1.8.0
at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorIm
at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAc
at java.lang.reflect.Constructor.newInstance(Constructor.java:423) ~[na:1.8.0_171]
at com.mysql.cj.exceptions.ExceptionFactory.createException(ExceptionFactory.java:59
at com.mysql.cj.exceptions.ExceptionFactory.createException(ExceptionFactory.java:10
at com.mysql.cj.exceptions.ExceptionFactory.createException(ExceptionFactory.java:14
at com.mysql.cj.exceptions.ExceptionFactory.createCommunicationsException(ExceptionF
at com.mysql.cj.protocol.a.NativeProtocol.negotiateSSLConnection(NativeProtocol.java
at com.mysql.cj.protocol.a.NativeAuthenticationProvider.negotiateSSLConnection(Nativ
at com.mysql.cj.protocol.a.NativeAuthenticationProvider.proceedHandshakeWithPluggabl
at com.mysql.cj.protocol.a.NativeAuthenticationProvider.connect(NativeAuthentication
```

```

at com.mysql.cj.protocol.a.NativeProtocol.connect (NativeProtocol.java:1411) ~[mysql-
at com.mysql.cj.NativeSession.connect (NativeSession.java:165) ~[mysql-connector-java
at com.mysql.cj.jdbc.ConnectionImpl.connectWithRetries (ConnectionImpl.java:876) ~[my
... 35 common frames omitted
Caused by: javax.net.ssl.SSLHandshakeException: java.security.cert.CertificateExcepti
at sun.security.ssl.Alerts.getSSLException (Alerts.java:192) ~[na:1.8.0_171]
at sun.security.ssl.SSLSocketImpl.fatal (SSLSocketImpl.java:1964) ~[na:1.8.0_171]
at sun.security.ssl.Handshaker.fatalSE (Handshaker.java:328) ~[na:1.8.0_171]
at sun.security.ssl.Handshaker.fatalSE (Handshaker.java:322) ~[na:1.8.0_171]
at sun.security.ssl.ClientHandshaker.serverCertificate (ClientHandshaker.java:1614) ~
at sun.security.ssl.ClientHandshaker.processMessage (ClientHandshaker.java:216) ~[na:
at sun.security.ssl.Handshaker.processLoop (Handshaker.java:1052) ~[na:1.8.0_171]
at sun.security.ssl.Handshaker.process_record (Handshaker.java:987) ~[na:1.8.0_171]
at sun.security.ssl.SSLSocketImpl.readRecord (SSLSocketImpl.java:1072) ~[na:1.8.0_171]
at sun.security.ssl.SSLSocketImpl.performInitialHandshake (SSLSocketImpl.java:1385) ~
at sun.security.ssl.SSLSocketImpl.startHandshake (SSLSocketImpl.java:1413) ~[na:1.8.0
at sun.security.ssl.SSLSocketImpl.startHandshake (SSLSocketImpl.java:1397) ~[na:1.8.0
at com.mysql.cj.protocol.ExportControlled.performTlsHandshake (ExportControlled.java:
at com.mysql.cj.protocol.StandardSocketFactory.performTlsHandshake (StandardSocketFac
at com.mysql.cj.protocol.a.NativeSocketConnection.performTlsHandshake (NativeSocketCo
at com.mysql.cj.protocol.a.NativeProtocol.negotiateSSLConnection (NativeProtocol.java
... 41 common frames omitted
Caused by: java.security.cert.CertificateException: java.security.cert.CertPathValida
at com.mysql.cj.protocol.ExportControlled$X509TrustManagerWrapper.checkServerTrusted
at sun.security.ssl.AbstractTrustManagerWrapper.checkServerTrusted (SSLContextImpl.ja
at sun.security.ssl.ClientHandshaker.serverCertificate (ClientHandshaker.java:1596) ~
... 52 common frames omitted
Caused by: java.security.cert.CertPathValidatorException: Path does not chain with an
at sun.security.provider.certpath.PKIXCertPathValidator.validate (PKIXCertPathValidat
at sun.security.provider.certpath.PKIXCertPathValidator.engineValidate (PKIXCertPathV
at java.security.cert.CertPathValidator.validate (CertPathValidator.java:292) ~[na:1.
at com.mysql.cj.protocol.ExportControlled$X509TrustManagerWrapper.checkServerTrusted
... 54 common frames omitted

```

You have two options:

- Configure SSL for the MySQL connection.
- Switch from "**Strong Password Encryption for Authentication**" to "**Legacy Authentication Method**".



Some url what might be useful for you is you decide going ahead with SSL configuration:

- <https://dev.mysql.com/doc/connector-j/8.0/en/connector-j-reference-using-ssl.html>
- <https://dev.mysql.com/doc/refman/8.0/en/creating-ssl-rsa-files-using-mysql.html>

In Windows is quite easy to enable "**Legacy Authentication Method**":

- Execute the installer.
- Click on reconfigure beside MySQL Server.
- Follow the wizard screens and choose "**Legacy Authentication Method**".

Old JDBC driver

If in your log file is shown the error below:


```
Caused by: java.lang.NullPointerException: null
    at com.mysql.jdbc.ConnectionImpl.getServerCharset (ConnectionImpl.java:3005) ~[mysql-connector-java-5.1.38-bin.jar:5.1.38]
    at com.mysql.jdbc.MysqlIO.sendConnectionAttributes (MysqlIO.java:1916) ~[mysql-connector-java-5.1.38-bin.jar:5.1.38]
    at com.mysql.jdbc.MysqlIO.proceedHandshakeWithPluggableAuthentication (MysqlIO.java:1871) ~[mysql-connector-java-5.1.38-bin.jar:5.1.38]
    at com.mysql.jdbc.MysqlIO.doHandshake (MysqlIO.java:1215) ~[mysql-connector-java-5.1.38-bin.jar:5.1.38]
    at com.mysql.jdbc.ConnectionImpl.coreConnect (ConnectionImpl.java:2255) ~[mysql-connector-java-5.1.38-bin.jar:5.1.38]
    at com.mysql.jdbc.ConnectionImpl.connectWithRetries (ConnectionImpl.java:2106) ~[mysql-connector-java-5.1.38-bin.jar:5.1.38]
    ... 41 common frames omitted
```

You are using a wrong driver version for connecting to the MySQL Server 8, you should make the next changes:

- Replace the `mysql-connector-java-5.1.38-bin.jar` into your `$TOMCAT_HOME/libs` by the latest JDBC driver for MySQL Server 8. You can get the latest driver from <https://dev.mysql.com/downloads/connector/j/>.
- Change the MySQL connection settings set into your `$TOMCAT_HOME/conf/server.xml`, should be something like:



In the latest version of MySQL JDBC driver, the name of the class **`com.mysql.jdbc.Driver`** has been changed to **`com.mysql.cj.jdbc.Driver`**.

```
<Resource name="jdbc/OpenKMDS" auth="Container" type="javax.sql.DataSource"
    maxTotal="100" maxIdle="30" maxWaitMillis="10000" validationQuery="select 1"
    username="openkm" password="openkm" driverClassName="com.mysql.cj.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/okmdb?autoReconnect=true&useUnicode=true&characterEncoding=utf8"/>
```

The server time zone value is unrecognized

If in your log file is shown the error below or similar:

```
Caused by: com.mysql.cj.exceptions.InvalidConnectionAttributeException: The server time zone value 'UTC' is unrecognized
    at sun.reflect.NativeConstructorAccessorImpl.newInstance0 (Native Method) ~[na:1.8.0_171]
    at sun.reflect.NativeConstructorAccessorImpl.newInstance (NativeConstructorAccessorImpl.java:62) ~[na:1.8.0_171]
    at sun.reflect.DelegatingConstructorAccessorImpl.newInstance (DelegatingConstructorAccessorImpl.java:46) ~[na:1.8.0_171]
    at java.lang.reflect.Constructor.newInstance (Constructor.java:423) ~[na:1.8.0_171]
    at com.mysql.cj.exceptions.ExceptionFactory.createException (ExceptionFactory.java:59) ~[mysql-connector-java-8.0.29.jar:8.0.29]
    at com.mysql.cj.exceptions.ExceptionFactory.createException (ExceptionFactory.java:83) ~[mysql-connector-java-8.0.29.jar:8.0.29]
    at com.mysql.cj.util.TimeUtil.getCanonicalTimezone (TimeUtil.java:128) ~[mysql-connector-java-8.0.29.jar:8.0.29]
    at com.mysql.cj.protocol.a.NativeProtocol.configureTimezone (NativeProtocol.java:220) ~[mysql-connector-java-8.0.29.jar:8.0.29]
    at com.mysql.cj.protocol.a.NativeProtocol.initServerSession (NativeProtocol.java:222) ~[mysql-connector-java-8.0.29.jar:8.0.29]
    at com.mysql.cj.jdbc.ConnectionImpl.initializePropsFromServer (ConnectionImpl.java:130) ~[mysql-connector-java-8.0.29.jar:8.0.29]
    at com.mysql.cj.jdbc.ConnectionImpl.connectWithRetries (ConnectionImpl.java:895) ~[mysql-connector-java-8.0.29.jar:8.0.29]
    ... 45 common frames omitted
```

It seems the latest MySQL JDBC driver have some issue with your OS time zone configuration, a quick workaround is to indicate the server time zone, for example **`serverTimezone=UTC`**:

```
<Resource name="jdbc/OpenKMDS" auth="Container" type="javax.sql.DataSource"
    maxTotal="100" maxIdle="30" maxWaitMillis="10000" validationQuery="select 1"
    username="openkm" password="openkm" driverClassName="com.mysql.cj.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/okmdb_test?autoReconnect=true&useUnicode=true&characterEncoding=utf8&serverTimezone=UTC"/>
```

Problem connecting to your database

If you get an error like this:

```
Caused by: com.mysql.jdbc.exceptions.jdbc4.MySQLNonTransientConnectionException: Publ
```

You need to modify your file **conf/server.xml** and add to URL by JDBC the follow: **allowPublicKeyRetrieval=true**.

```
<Resource name="jdbc/OpenKMDS" auth="Container" type="javax.sql.DataSource"
    maxTotal="100" maxIdle="30" maxWaitMillis="10000" validationQuery="select
    username="openkm" password="openkm" driverClassName="com.mysql.cj.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/okmdb_test?allowPublicKeyRetrieval=true&
```

Configuring Oracle

Database creation

Create tablespace

```
create tablespace OPENKMDATA datafile '/u01/app/oracle/oradata/XE/openkmdata.dbf' size 100M;
```

Create a database and user

```
create user OPENKM identified by *secret* default tablespace OPENKMDATA quota unlimited;
grant all privileges to OPENKM;
grant create any context, create cluster, create dimension, create indextype, create
```

Configure your OpenKM.cfg



This configuration property should be set before the database creation. Once the database has been initialized don't modify it because can damage your installation.

If your OpenKM installation has been already configured with another database you can't switch to another database simply changing this property.

Edit the file \$TOMCAT_HOME/OpenKM.cfg

```
hibernate.dialect=org.hibernate.dialect.Oracle10gDialect
hibernate.hbm2ddl=create
```

Configure Tomcat datasources

Edit the file \$TOMCAT_HOME/conf/server.xml and enable the resource named jdbc/OpenKMDS

```
<Resource name="jdbc/OpenKMDS" auth="Container" type="javax.sql.DataSource"
  maxActive="100" maxIdle="30" maxWait="10000" validationQuery="select 1 from dual"
  username="OPENKM" password="*secret*" driverClassName="oracle.jdbc.driver.OracleDriver"
  url="jdbc:oracle:thin:@localhost:1521:XE"/>
```



Configure the attributes values named:

- `userName`.
- `password`.
- `url` (change to your hosts and port).

Can be only one resource named `jdbc/OpenKMDS`.

Sample using service name instead of SID:



The main difference between SID connection is the ":" is replaced by "/".

```
<Resource name="jdbc/OpenKMDS" auth="Container" type="javax.sql.DataSource"
maxActive="100" maxIdle="30" maxWait="10000" validationQuery="select 1 from dual"
username="OPENKM" password="*secret*" driverClassName="oracle.jdbc.driver.OracleDriver"
url="jdbc:oracle:thin:@localhost:1521/ServiceName"/>
```

Configure application login

Edit the file \$TOMCAT_HOME/OpenKM.xml

```
<security:authentication-manager alias="authenticationManager">
  <security:authentication-provider>
    <security:password-encoder hash="md5"/>
    <security:jdbc-user-service
      data-source-ref="dataSource"
      users-by-username-query="select usr_id, usr_password, 1 from OKM_USER where u
      authorities-by-username-query="select ur_user, ur_role from OKM_USER_ROLE whe
    </security:authentication-provider>
  </security:authentication-manager>
```

Check for JDBC Driver

Go to \$TOMCAT_HOME/lib and check for JDBC **ojdbc6-11.2.0.3.jar** Driver.

If it is not present, download Oracle JDBC driver from [Oracle JDBC Driver page](#) and move it.



The driver numeration version could be distinct due some version upgrade.

Run application

As a process into OpenKM startup, the application will create automatically and empty database structure.



Once the tables are created, OpenKM will automatically change the **hibernate.hbm2ddl** property from **create** to **none**.

Configuring PostgreSQL

Database creation

Create a database and user

```
DROP DATABASE IF EXISTS okmdb;  
CREATE USER openkm WITH PASSWORD '*secret*';  
CREATE DATABASE okmdb WITH OWNER openkm ENCODING 'UTF8';
```

or from the command line

```
$ createuser --pwprompt openkm  
$ createdb --owner=openkm --encoding=UTF8 okmdb
```

More info at:

- [PostgreSQL 9.1 Documentation](#)
- [Guía Ubuntu: PostgreSQL](#)

Configure your OpenKM.cfg



This configuration property should be set before the database creation. Once the database has been initialized don't modify it because can damage your installation.

If your OpenKM installation has been already configured with another database you can't switch to another database simply changing this property.

Edit the file \$TOMCAT_HOME/OpenKM.cfg

```
hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect  
hibernate.hbm2ddl=create
```

Configure Tomcat datasources

Edit the file \$TOMCAT_HOME/conf/server.xml and enable the resource named jdbc/OpenKMDS

```
<Resource name="jdbc/OpenKMDS" auth="Container" type="javax.sql.DataSource"  
    maxActive="100" maxIdle="30" maxWait="10000" validationQuery="select 1"  
    username="openkm" password="*secret*" driverClassName="org.postgresql.Driver"  
    url="jdbc:postgresql://localhost:5432/okmdb"/>
```



Configure the attributes values named:

- *userName.*
- *password.*

- url (change to your hosts and port).

Can be only one resource named **jdbc/OpenKMDS**.

Configure application login

Edit the file \$TOMCAT_HOME/OpenKM.xml

```
<security:authentication-manager alias="authenticationManager">
  <security:authentication-provider>
    <security:password-encoder hash="md5"/>
    <security:jdbc-user-service
      data-source-ref="dataSource"
      users-by-username-query="select usr_id, usr_password, 1 from OKM_USER where u
      authorities-by-username-query="select ur_user, ur_role from OKM_USER_ROLE whe
    </security:authentication-provider>
  </security:authentication-manager>
```

Check for JDBC Driver

Go to \$TOMCAT_HOME/lib and check for JDBC **postgresql-9.1-902.jdbc4.jar** Driver.

If is not present, download PostgreSQL JDBC driver from [PostgreSQL Driver Page](#) and move it.



The driver numeration version could be distinct due some version upgrade.

Run application

As a process into OpenKM startup, the application will create automatically and empty database structure.



Once the tables are created, OpenKM will automatically change the **hibernate.hbm2ddl** property from **create** to **none**.

Additional information

If you want to obtain the maximun performance from PostgreSQL, [pgFouine](#) will help in your task.

PostgreSQL 9.6 and upper

From **PostgreSQL version 9.6 and upper**, has been **changed some tables and columns of the schema**. Check that you are using the right JDBC driver.

You can find the current JDBC driver in \$TOMCAT_HOME/libs



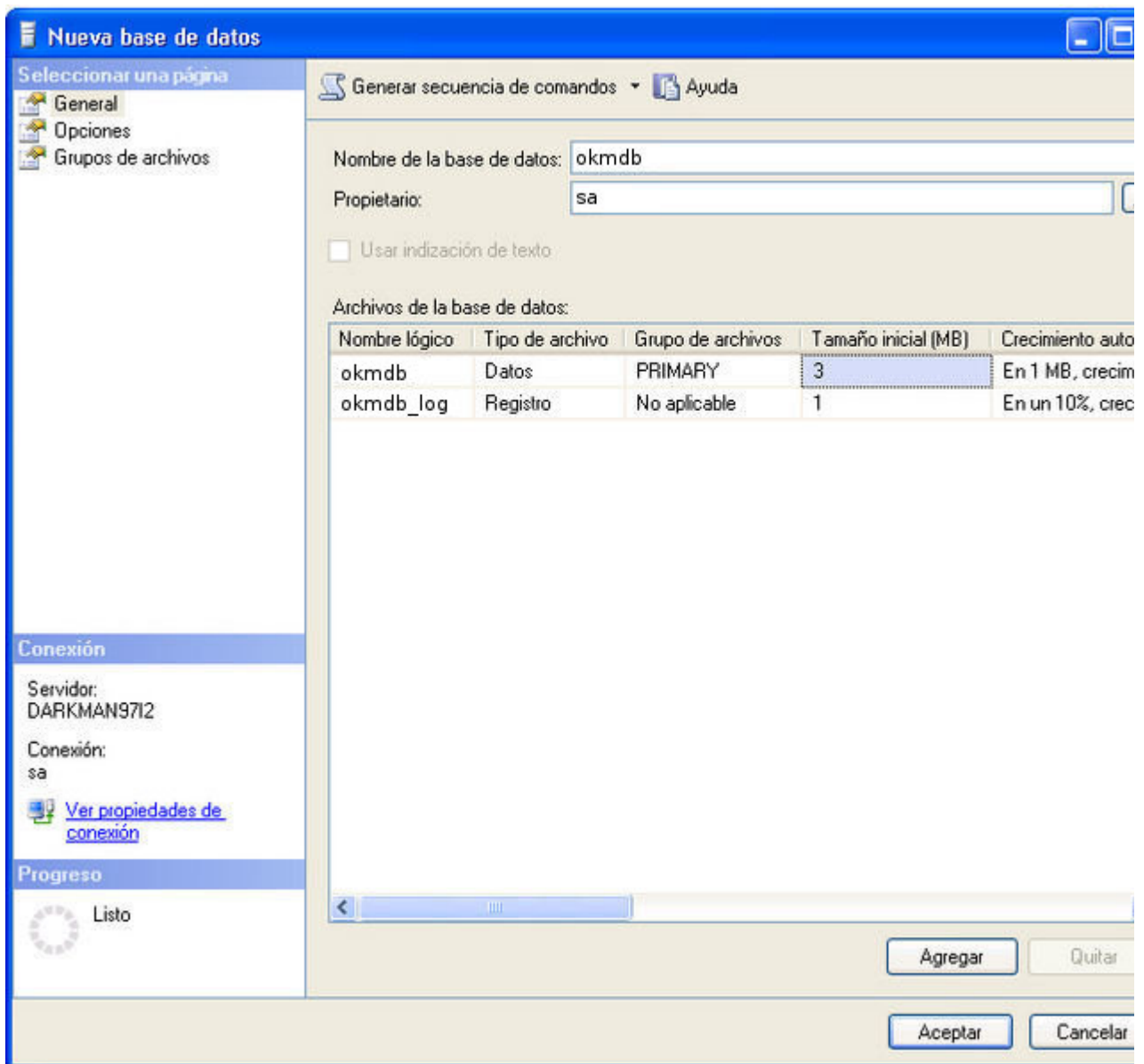
Read this related issue for more information [PostgreSQL - ERROR o.h.tool.hbm2ddl.SchemaUpdate - could not complete schema update](#)

Configuring SQL Server

Database creation

Create a database and user

[Start SQL Server Management Studio Express](#) and create a database called okmdb.



Select your appropriate database collate.

Nueva base de datos

Seleccionar una página

- General
- Opciones
- Grupos de archivos

Generar secuencia de comandos Ayuda

Intercalación: Modern_Spanish_CI_AI_KS

Modelo de recuperación: Simple

Nivel de compatibilidad: SQL Server 2005 (90)

Otras opciones:

Automático

Actualizar estadísticas automática y asíncrona	False
Actualizar estadísticas automáticamente	True
Cerrar automáticamente	False
Crear estadísticas automáticamente	True
Reducir automáticamente	False

Cursor

Cierre de cursor al confirmar habilitado	False
Cursor predeterminado	GLOBAL

Estado

Base de datos de sólo lectura	False
Estado de base de datos	NORMAL
Restringir acceso	MULTI_USER

Recuperación

Verificación de páginas	CHECKSUM
-------------------------	----------

Varios

Advertencias ANSI habilitadas	False
Anulación aritmética habilitada	False
Anulación exacta numérica	False

Actualizar estadísticas automática y asíncronamente

Conexión

Servidor: DARKMAN9712

Conexión: sa

[Ver propiedades de conexión](#)

Progreso

Listo

Aceptar Ca



We checked with *Modern_Spanish_CI_AI_KS*, where *CI* indicate *case-insensitive* collation and *AS* indicate *accent-sensitive* collation.

If you're not sure about your collation could also try with *SQL_Latin1_General_CP1_CI_AS*.

Configure your OpenKM.cfg



This configuration property should be set before the database creation. Once the database has been initialized don't modify it because can damage your installation.

If your OpenKM installation has been already configured with another database you can't switch to another database simply changing this property.

Edit the file \$TOMCAT_HOME/OpenKM.cfg

```
hibernate.dialect=org.hibernate.dialect.SQLServerDialect
hibernate.hbm2ddl=create
```

Configure Tomcat data sources

Edit the file \$TOMCAT_HOME/conf/server.xml and enable the resource named jdbc/OpenKMDS

```
<Resource name="jdbc/OpenKMDS" auth="Container" type="javax.sql.DataSource"
    maxActive="100" maxIdle="30" maxWait="10000" validationQuery="select 1"
    username="sa" password="***" driverClassName="com.microsoft.sqlserver.jdbc.
    url="jdbc:sqlserver://localhost:1433;databaseName=okmdb"/>
```



Configure the attributes values named:

- `userName`.
- `password`.
- URL (change to your hosts and port).

Can be only one resource named `jdbc/OpenKMDS`.

Configure application login

Edit the file \$TOMCAT_HOME/OpenKM.xml

```
<security:authentication-manager alias="authenticationManager">
  <security:authentication-provider>
    <security:password-encoder hash="md5"/>
    <security:jdbc-user-service
      data-source-ref="dataSource"
      users-by-username-query="select usr_id, usr_password, 1 from OKM_USER where u.
      authorities-by-username-query="select ur_user, ur_role from OKM_USER_ROLE whe.
    </security:authentication-provider>
  </security:authentication-manager>
```

Check for JDBC Driver

Go to \$TOMCAT_HOME/lib and check for JDBC `sqljdbc4.jar` Driver.

If it is not present, download SQL Server JDBC driver from [Microsoft SQL Server Driver Page](#) and move it.



The driver numeration version could be distinct due to some version upgrade.

Other JDBC driver for MS SQL Server we've tested is [jdts](#).

Run application

As a process into the OpenKM startup, the application will create automatically an empty database structure.



Once the tables are created, OpenKM will automatically change the **hibernate.hbm2ddl** property from **create** to **none**.

Troubleshooting

You can not connect to MS SQL Server

Ensure you've got TCP protocol enabled (by default is disabled)



Connection lost after some time

We have observed in some SQL Server editions what seems a new behaviour what passed some hours the connection is closed (might be some extra configuration parameter in the latest SQL server editions what comes enabled by default or similar).

There's a new configuration attribute name **autoReconnect** what solves it.

```
<Resource name="jdbc/OpenKMDS" auth="Container" type="javax.sql.DataSource"
maxActive="100" maxIdle="30" maxWait="10000" validationQuery="select 1"
username="sa" password="****" driverClassName="com.microsoft.sqlserver.jdbc.
url="jdbc:sqlserver://localhost\SQL;databaseName=okmdb;autoReconnect=true"/
```

Not able to login into OpenKM

When first-time OpenKM login it fails with right username and password (**okmAdmin** / **admin**), you should check database configuration.

Check if the database is configured to be **case-sensitive**, might be one of the reasons why you are not able to login.



*If you have this troubleshooting, you should to drop the database and create again selecting the collation as **case-insensitive**.*

Additional information

- [*How to enable remote connections in SQL Server 2008*](#)
- [*How do I configure SQL Server 2005/2008 to accept SQL Authentication*](#)

Configuring other databases

OpenKM uses Hibernate that means can be installed on major DBMS. For more information take a look at [Hibernate: SQL Dialects](#).

By default application has been tested in:

- MySQL or MariaDB.
- PostgreSQL.
- MS SQL Server.
- Oracle.



If you want to check other database, ask us at our [public forum](#).

Backup and restore

One of the most important things to have is an updated backup of your OpenKM.

If you have followed the installation instructions then you have configured OpenKM to use a production-safe database and optionally you have set a path the OpenKM repository data store.



OpenKM has several configurations, but for optimal performance we recommend:

- Use the default data store that saves binary information directly to the server disk (in a non human comprehensible format).
- All the other information, folders and documents structure, metadata, etc., is stored in the database.



OpenKM allows the use of other datastores, for example to save binary data into a database.

In this case:

1. First backup your database (make a SQL dump)
2. Then backup your data on filesystem



By default, all documents are stored in `$TOMCAT_HOME/repository`. This default folder can be changed using the property **repository.home** in the file `$TOMCAT_HOME/OpenKM.cfg` file.



Although OpenKM supports hot backups of your data, it is recommended to stop Tomcat before starting the backup. Apply this good practice when possible.

Backup steps:

- Stop Tomcat (optional).
- Backup the database (mandatory).
- Backup the repository folder (mandatory).
- Start Tomcat (optional).



Performing the backup in that order (the database first, then the filesystem) will ensure backup consistency.

There is an alternative to traditional backup methods: you can export the whole repository from OpenKM Administration > Utilities > [Repository export](#) and choose metadata and history options. But we do not suggest this way of doing backups. For medium to large repositories we suggest some incremental backup solution.

Detailed steps

In case you have OpenKM configured with MySQL or MariaDB you can follow these steps to backup and restore. For other databases check its corresponding command.

Backup

```
$ /etc/init.d/tomcat stop  
$ mysqldump -h localhost -u root -p okmdb > /path/to/openkm/okmdb.sql  
$ zip -r /path/to/backup/okm-backup.zip /path/to/openkm  
$ /etc/init.d/tomcat start
```

After that, you will have a ZIP archive with the whole OpenKM installation and the exported database. Please, check before destroying the installation. We always recommend to restore the backup to ensure you have all the required files.

Restore

Please, note that you have to create the database if wasn't created previously. In case of MySQL or MariaDB install it:

```
$ sudo apt install mariadb-server
```

And execute the following sentences:

```
CREATE USER 'openkm'@'localhost' IDENTIFIED BY 's3cr3t0';  
CREATE DATABASE okmdb DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_bin;  
GRANT ALL ON okmdb.* TO 'openkm'@'localhost' WITH GRANT OPTION;
```

Keep on mind that the user and database name may be different in you case. Please, take a look at the installation report or check it at \$TOMCAT_HOME/conf/server.xml

Once created restore the database and files:

```
$ unzip /path/to/backup/okm-backup.zip -d /path/to/restore  
$ mysql -h localhost -u root -p okmdb < /path/to/restore/okmdb.sql
```

Set the permissions according to the user who start OpenKM:

```
$ sudo chown openkm:openkm /path/to/openkm -R
```

Install the required dependencies:

```
$ sudo apt install gawk imagemagick libreoffice tesseract-ocr
```

Once restored, verify file permissions, start OpenKM and make a repository check. See [Repository checker](#).

Backup scripts

- [*Linux backup and restore with rdiff-backup tool*](#) [*Linux backup and restore with rdiff-backup tool*](#)
- [*Linux backup and restore with rsync*](#)
- [*Linux backup and restore with Duplicity*](#)
- [*Linux backup with LFTP*](#)

Additional information

- [*Rdiff-backup home page*](#)
- [*Easy Automated Snapshot-Style Backups with Linux and Rsync*](#)
- [*Duplicity Backup Howto*](#)
- [*DAR differential backup mini-howto*](#).

Linux backup and restore with BorgBackup

Borg is a deduplicating backup program which also supports compression and authenticated encryption. It provides an efficient and secure way to backup data. The data deduplication technique used makes Borg suitable for daily backups since only changes are stored. The authenticated encryption technique makes it suitable for backups to not fully trusted targets.

A more detailed description of BorgBackup can be found at <https://borgbackup.readthedocs.io/en/stable/index.html>.

Installation

Borg is included in many Linux distributions repositories but sometimes the packaged version is quite old, and it is better to install manually. This installation is as simple as:



In the official borg releases website section you will find the latest version:

<https://github.com/borgbackup/borg/releases>

```
$ wget https://github.com/borgbackup/borg/releases/download/1.1.5/borg-linux64
$ chmod +x borg-linux64
$ mv borg-linux64 /usr/local/bin/borg
```

Sample script

Backup to remote computer with SSH

```
#!/bin/bash
#
## BEGIN CONFIG ##
HOST=$(uname -n)
OPENKM_DB="okmdb"
OPENKM_HOME="/home/openkm"
TOMCAT_HOME="$OPENKM_HOME/tomcat"
DATABASE_EXP="$OPENKM_HOME/db"
BACKUP_HOST="backup@bk-server.com"
BACKUP_DIR="/mnt/backup/$HOST"
BACKUP_REPO="$${BACKUP_HOST}:${BACKUP_DIR}"
## END CONFIG ##

# Check root user
if [ $(id -u) != 0 ]; then echo "You should run this script as root"; exit; fi

echo "Backup destination ${BACKUP_REPO}"
if (ssh ${BACKUP_HOST} "[ ! -d ${BACKUP_DIR} ]"); then
    echo "Initialize backup repository: ${BACKUP_DIR}"
    ssh ${BACKUP_HOST} "mkdir -p ${BACKUP_DIR}"
    borg init -e none ${BACKUP_REPO}
fi

# Database
echo "Backup MySQL databases"
mkdir -p ${DATABASE_EXP}
echo "-- Backeping $OPENKM_DB..."
mysqldump --single-transaction --add-drop-table --quick ${OPENKM_DB} > ${DATABASE_EXP}
```



```
borg create -v --stats ${BACKUP_REPO}::'now:%Y-%m-%d_%H:%M' ${OPENKM_HOME}
borg prune -v ${BACKUP_REPO} --list --keep-daily=7 --keep-weekly=4 --keep-monthly=6
```

This is an improved SSH version

```
#!/bin/bash

# Configuration
DATABASE_EXP="/home/openkm/db"
BACKUP_HOST="backup@bk-server.com"
BACKUP_DIR="/mnt/backup/${HOST}"
BACKUP_REPO="${BACKUP_HOST}:${BACKUP_DIR}"

echo "### BEGIN: $(date +%x %X)" ###
echo "Backup destination ${BACKUP_REPO}"
TBEGIN=$(date +%s")

# Check if the repository is already initialized
borg list ${BACKUP_REPO} > /dev/null
RESULT=$?; if [[ ${RESULT} != 0 ]]; then
    echo "Initializing backup repository..."
    borg init -e none ${BACKUP_REPO}
fi

# Database
echo "Backup MySQL databases"
mkdir -p ${DATABASE_EXP}
MYSQL_DBS=$(mysqlshow | awk '(NR > 2) && ([a-zA-Z0-9]+[ ]+[ ])/) { print $2 }');
for DB in $MYSQL_DBS; do
    if [[ $DB != "mysql" && $DB != "test" && $DB != "information_schema" && $DB != "performance_schema" ]]; then
        echo "- Backing up $DB..."
        mysqldump --single-transaction --add-drop-table --quick ${DB} > ${DATABASE_EXP}/${DB}.sql
    fi
done

borg create -v --stats ${BACKUP_REPO}::'now:%Y-%m-%d_%H:%M' /opt/var/etc/root /h
RESULT=$?; if [[ ${RESULT} != 0 ]]; then
    echo "*****"
    echo "Backup error => exit code: ${RESULT}"
    echo "*****"
fi

# Prune old backups
echo "Prune old backups"
borg prune -v --list --keep-within=30d ${BACKUP_REPO}
RESULT=$?; if [[ ${RESULT} != 0 ]]; then
    echo "*****"
    echo "Backup error => exit code: ${RESULT}"
    echo "*****"
fi

TEND=$(date +%s")
TDIFF=$(( $TEND - $TBEGIN ))
echo "### END: $(date +%x %X)" - $(( $TDIFF / 60 )) mins and $(( $TDIFF % 60 )) secs ###
```

Backup to mapped network

```
#!/bin/bash
#
## BEGIN CONFIG ##
HOST=$(uname -n)
```

```

OPENKM_DB="okmdb"
OPENKM_HOME="/home/openkm"
TOMCAT_HOME="$OPENKM_HOME/tomcat"
DATABASE_EXP="$OPENKM_HOME/db"
BACKUP_DIR="/mnt/backup"
BACKUP_REPO="${BACKUP_DIR}/${HOST}"
## END CONFIG ##

# Check root user
if [ $(id -u) != 0 ]; then echo "You should run this script as root"; exit; fi

# Mount disk
if mount | grep "$BACKUP_DIR type" > /dev/null; then
    echo "$BACKUP_DIR already mounted";
else
    mount "$BACKUP_DIR";

    if mount | grep "$BACKUP_DIR type" > /dev/null; then
        echo "$BACKUP_DIR mounted";
    else
        echo "$BACKUP_DIR error mounting";
        exit -1;
    fi
fi

# Initialize repository
echo "Backup destination ${BACKUP_REPO}"
if ! ls -l ${BACKUP_DIR} | grep '^d' | grep "${HOST}" > /dev/null; then
    echo "Initialize backup repository: ${BACKUP_REPO}"
    mkdir ${BACKUP_REPO}
    borg init -e none ${BACKUP_REPO}
fi

# Database
echo "Backup MySQL databases"
mkdir -p ${DATABASE_EXP}
echo "- Backuping $OPENKM_DB..."
mysqldump --single-transaction --add-drop-table --quick ${OPENKM_DB} > ${DATABASE_EXP}

# Data backup
echo "File system backup: ${BACKUP_REPO}"
borg create -v --stats ${BACKUP_REPO}::'{now:%Y-%m-%d_%H:%M}' ${OPENKM_HOME}
borg prune -v ${BACKUP_REPO} --list --keep-daily=7 --keep-weekly=4 --keep-monthly=6

# Umount disk
sync
umount "$BACKUP_DIR"
echo "$BACKUP_DIR umounted";

```

Configure crontab

To install the cron job, run:

```

$ sudo mkdir /root/logs

$ sudo crontab -e

```

And add these lines according to your personal configuration:

```

MAILTO=nomail@openkm.com
PATH=/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/bin
@daily /root/backup.sh 2>&1 | tee /root/logs/backup.$(date +%Y.%m.%d_%H.%M.%S)

```



More information at [Crontab quick reference](#)

If you want to be notified by mail you should install "**postfix**" service in your server.

Recovery

In order to list all stored backup use the **list** action parameter:

```
$ borg list backup@bk-server.com:/mnt/backup/name
2018-02-07 Thu, 2018-02-07 10:12:31 [c878b314107c76b2d86ba75a9e5b2a216bf1f0d6b824313396c51ccda02a72e6]
2018-02-08 Thu, 2018-02-08 10:15:45 [2cf11f13bc7a2c9c0f1a0c243da52ac4b7657e7c6429cca0bf06a80467837f1e]
```

If you want to list all files in a given backup:

```
$ borg list backup@bk-server.com:/mnt/backup/name::2018-02-07_09:27
```

In case you need to access the contents of a backup you can mount as a local filesystem:

```
$ mkdir /tmp/backup
$ borg mount backup@bk-server.com:/mnt/backup/name::2018-02-07_09:27 /tmp/backup
```



If you do not specify an specific backup then will be mounted all the backup. The command line should be:

```
$ mkdir /tmp/backup
$ borg mount backup@bk-server.com:/mnt/backup /tmp/backup
```

At the end you might be interested in unmount the folder

```
borg umount /tmp/backup
```

Linux backup and restore with rdiff-backup tool

Rdiff-backup backs up one directory to another, possibly over a network. The target directory ends up a copy of the source directory, but extra reverse diffs are stored in a special subdirectory of that target directory, so you can still recover files lost some time ago.

The idea is to combine the best features of a mirror and an incremental backup. rdiff-backup also preserves subdirectories, hard links, dev files, permissions, uid/gid ownership, modification times, extended attributes, acls, and resource forks.

Also, rdiff-backup can operate in a bandwidth efficient manner over a pipe, like rsync. Thus you can use rdiff-backup and ssh to securely back a hard drive up to a remote location, and only the differences will be transmitted.



If you need to backup to CIFS (SMB, Samba) or Mac's HFS, please take a look at [rdiff-backup FAQ](#).

Debian & Ubuntu

This utility is included in the software repository, so to install just run as root:

```
$ apt-get install rdiff-backup
```



Rdiff-backup from Debian issues a warning message due to a deprecated method [os.popen2 is deprecated], which can be hidden following the steps at [How to shut up Python deprecation warnings](#).

CentOS & RedHat

In this case, the application is not included in the default software repository so you need to make use of the [EPEL](#) repository:

```
$ rpm -Uvh http://dl.fedoraproject.org/pub/epel/7/x86\_64/e/epel-release-7-5.noarch.rpm
$ yum update
$ yum install rdiff-backup
```

Preliminaries

In these scripts are assumed these default values:

- **\$TOMCAT_HOME** value is **"/home/openkm/tomcat"**
- OpenKM **database** is named **"okmdb"**.
- OpenKM **database password** value is **"*secret*"**.
- OpenKM start & stop **service script** location is **"/etc/init.d/tomcat"**

Sample script



As a good practice the backup should be done by root user.



The script below does a backup to USB and the application database is MySQL.

USB disk mount point can be defined in `/etc/fstab` as:



```
/dev/sdb1    /mnt/backup    ext4    defaults    0    0
```

These are the global script sections:

- Global var configuration under **## BEGIN CONFIG ##** section.
- Checking user root under **# Check root** user section.
- Deleting older database backup **# Delete older local database backup** section.
- Mount disk (optional section) under **# Mount disk** section.
- Stop application (optional section) under **# Stop tomcat** section.
- Clean logs (optional section) under **# Clean logs** section.
- Backup the database to the filesystem under **#Backup database** section.
- Backup repository and database dump to backup destination under **#Backup and purge old backups** section.
- Start application (optional section) **# Start tomcat** section.
- Show statistics (optional section) **# Status** section.
- Umount disk (optional section) under **# Umount disk** section.

Configuration parameters explanation:

Parameter	Description
HOST	The server host name.
DATABASE_PASS	The database password. <div> <i>The dataase user to do backup is always root user.</i> </div>
OPENKM_DB	The application database name. <div> <i>Usually will be named "okmdb".</i> </div>

OPENKM_HOME	The openkm home folder.
TOMCAT_HOME	The tomcat home folder. <div>  Usually into OPENKM_HOME folder. </div>
DATABASE_EXP	The database dump folder. <div>  Usually into the same server. </div>
BACKUP_DIR	Backup destination. In case of a remote server, the form is user@server::/path/to/backup

Create a backup script **/root/backup.sh**

```
$ sudo su
```

```
$ vim /root/backup.sh
```

```
#!/bin/bash
#
## BEGIN CONFIG ##
HOST=$(uname -n)
DATABASE_PASS="*secret*"
OPENKM_DB="okmdb"
OPENKM_HOME="/home/openkm"
TOMCAT_HOME="$OPENKM_HOME/tomcat"
DATABASE_EXP="$OPENKM_HOME/db"
BACKUP_DIR="/mnt/backup"
## END CONFIG ##

# Check root user
if [ $(id -u) != 0 ]; then echo "You should run this script as root"; exit; fi

# Delete older local database backup
echo -e "### BEGIN: $(date +"%x %X") ###\n"
rm -rf $DATABASE_EXP
mkdir -p $DATABASE_EXP

# Mount disk
if mount | grep "$BACKUP_DIR type" > /dev/null; then
    echo "$BACKUP_DIR already mounted";
else
    mount "$BACKUP_DIR";

    if mount | grep "$BACKUP_DIR type" > /dev/null; then
        echo "$BACKUP_DIR mounted";
    else
        echo "$BACKUP_DIR error mounting";
        exit -1;
    fi
fi
```

```

# Stop Tomcat
/etc/init.d/tomcat stop

# Clean logs
#echo "Clean Tomcat temporal files."
#rm -rf $TOMCAT_HOME/logs/*
#rm -rf $TOMCAT_HOME/temp/*
#rm -rf $TOMCAT_HOME/work/Catalina/localhost

# Backup database
if [ -n "$DATABASE_PASS" ]; then
    echo "* Backeping MySQL data from $OPENKM_DB..."
    mysqldump -h localhost -u root -p$DATABASE_PASS $OPENKM_DB > $DATABASE_EXP/mysql_$O
    echo "-----";
fi

# Backup and purge old backups
rdiff-backup --remove-older-than 30B $BACKUP_DIR/$HOST
rdiff-backup -v 3 --print-statistics --include $OPENKM_HOME --exclude '*' / $BACKUP_

# Start Tomcat
/etc/init.d/tomcat start
echo -e "\n### END: $(date +%x %X)" ###"

# Status
echo "=====";
rdiff-backup --list-increment-sizes $BACKUP_DIR/$HOST
echo "*****";
df -h | grep "$BACKUP_DIR"
echo "=====";

# Umount disk
sync
umount "$BACKUP_DIR"

```

Increase the backup period

Modify the **# Backup and purge old backups** section increasing the parameter value:



The "30B" value indicate that will be able to restore a full backup of any of the last 30 days, increasing to "90B" will indicate any of the last 90 days, etc...

```
--remove-older-than 90B
```

Do remote backup

Modify the **# Backup and purge old backups** section.



The [backup@server](#) indicate remote "server" accessed by user "backup".

```

# Backup and purge old backups
rdiff-backup --remove-older-than 30B backup@server::$BACKUP_DIR/$HOST
rdiff-backup -v 3 --print-statistics --include $FILES --exclude '*' / backup@server:

```



Also could be only modified the **## Config section**, for example:

```
## BEGIN CONFIG ##
BACKUP_DIR="user@server::/path/to/backup"
```

And keep the **# Backup and purge old backups** intact.

```
# Backup and purge old backups
rdiff-backup --remove-older-than 30B $BACKUP_DIR/$HOST
```

Do remote backup to SMB or CIFS

Modify the **# Mount disk** section.

```
# Mount disk
if mount | grep "$BACKUP_DIR type" > /dev/null; then
    echo "$BACKUP_DIR already mounted";
else
    echo "Mounting $BACKUP_DIR ...";
    mount -t cifs //REMOTE_SERVER/REMOTE_DIR $BACKUP_DIR -o username=REMOTE_USER,password=REMOTE_PASS
fi

if mount | grep "$BACKUP_DIR type" > /dev/null; then
    echo "$BACKUP_DIR mounted";
else
    echo "Error mounting $BACKUP_DIR.";
    exit -1;
fi
fi
```

Do PostgreSQL backup

Modify the **# Backup database** section

```
# Backup de PostgreSQL
echo "* Backeping PostgreSQL data from $OPENKM_DB..."
su postgres -c "pg_dump $OPENKM_DB" > $DATABASE_EXP/pg_$OPENKM_DB.sql

# Databases optimizations
su postgres -c "vacuumdb -a -z" > /dev/null
su postgres -c "reindexdb -a -q" 2> /dev/null
```

Configure crontab

To install the cron job, run:

```
$ sudo mkdir /root/logs
$ sudo crontab -e
```

And add these lines according to your personal configuration:


```
MAILTO=nomail@openkm.com
@daily /root/backup.sh | tee /root/logs/backup.$(date +%Y.%m.%d_%H.%M.%S).log
```



More information at [Crontab quick reference](#)

If you want to be notified by mail you should install "**postfix**" service in your server.

Restoring backup

You can list the available backups:

```
$ rdiff-backup --list-increment-sizes /path/to/backup
Sun Sep 11 05:00:19 2012    4.25 GB    4.25 GB (current mirror)
Sun Sep  4 00:00:18 2012    13.3 MB    4.26 GB
Sun Aug 28 00:00:13 2012     674 MB    4.92 GB
Sun Aug 21 00:00:14 2012     5.50 MB    4.93 GB
Sun Aug 14 00:00:16 2012     1.75 MB    4.93 GB
Sun Aug  7 00:00:12 2012     288 KB    4.93 GB
Sun Jul 31 00:00:13 2012     43.0 KB    4.93 GB
Fri Jul 29 10:36:39 2012     5.56 KB    4.93 GB
```

Then decide to restore some of the backups, for example the one made on Sun Aug 28 00:00:13 2012:

```
$ rdiff-backup --restore-as-of 2012-08-28 /path/to/backup /path/to/destination
```

Or restore the last backup

```
$ rdiff-backup --restore-as-of now /path/to/backup /path/to/destination
```



The **--restore-as-of** parameter accepts several formats. See **rdiff-backup** documentation for more info.

Sample for restoring single folder (the `/home/openkm/tomcat/` must exists)

```
rdiff-backup --restore-as-of now /mnt/backup/ms1/home/openkm/tomcat/ /home/openkm/tomcat/
```



Inside of **/path/to/destination** you should see a directory **/home/openkm**, and inside it a couple of directories:

- **db**: The backup of the database.
- **tomcat-7.0.61**: The backup of the Tomcat installation and OpenKM repository into.

Additional information

- [rdiff-backup home page](#)

- [*rdiff-backup wiki*](#)
- [*Automated backups with rdiff-backup*](#)
- [*Using rdiff-backup for remote backups*](#)
- [*Suppress socket warnings when using rdiff-backup*](#)

Linux Sample Backup with Duplicity

Duplicity backs directories by producing encrypted tar-format volumes and uploading them to a remote or local file server.

Because duplicity uses librsync, the incremental archives are space efficient and only record the parts of files that have changed since the last backup. Because duplicity uses GnuPG to encrypt and/or sign these archives, they will be safe from spying and/or modification by the server.

Installation

Duplicity can be installed in Debian / Ubuntu as simple as:

```
$ sudo apt-get install duplicity ncftp python-boto python-paramiko
```



To avoid the **Import of duplicity.backends.giobackend Failed: No module named gio** warning, you need to install the **python-gobject-2** package:

```
$ apt-get install python-gobject-2
```

But it not in the CentOS / RedHat default repositories, so you need to install from another source. This script will help in this installation process:

```
#!/bin/bash

wget http://dl.fedoraproject.org/pub/epel/6/x86_64/duplicity-0.6.18-1.el6.x86_64.rpm
wget http://dl.fedoraproject.org/pub/epel/6/x86_64/ncftp-3.2.4-1.el6.x86_64.rpm
wget http://dl.fedoraproject.org/pub/epel/6/x86_64/librsync-0.9.7-15.el6.x86_64.rpm
wget http://dl.fedoraproject.org/pub/epel/6/x86_64/python-GnuPGInterface-0.3.2-6.el6..
wget http://dl.fedoraproject.org/pub/epel/6/x86_64/python-boto-2.5.2-1.el6.noarch.rpm

rpm -Uvh duplicity-0.6.18-1.el6.x86_64.rpm ncftp-3.2.4-1.el6.x86_64.rpm librsync-0.9.
```

Command line basics

To make a backup you can do:

```
$ duplicity --no-encryption /home/openkm file:///path/to/backup
```

To verify it:

```
$ duplicity --no-encryption verify file:///path/to/backup /home/openkm
```

To list backed files:

```
$ duplicity --no-encryption list-current-files file:///path/to/backup
```

To restore a single file:

```
$ duplicity --no-encryption --file-to-restore tomcat/OpenKM.cfg file:///path/to/backup RestoredBackup
```



The path to the file that is to be restored is related to the directory on which the backup set is based. So in the command above, `tomcat/OpenKM.cfg` plus the directory on which we based our backup (`/home/openkm`) equals `/home/openkm/tomcat/OpenKM.cfg`

Preliminaries

In these scripts are assumed these default values:

- `$TOMCAT_HOME` value is `"/home/openkm/tomcat"`
- OpenKM **database** is named `"okmdb"`.
- OpenKM **database password** value is `"*secret*"`.
- OpenKM start & stop **service script** location is `"/etc/init.d/tomcat"`

Sample script



As a good practice the backup should be done by root user.



The script below does a backup to USB and the application database is MySQL.

USB disk mount point can be defined in `/etc/fstab` as:

```
/dev/sdb1    /mnt/backup    ext4    defaults    0    0
```

These are the global script sections:

- Global var configuration under `## BEGIN CONFIG ##` section.
- Checking user root under `# Check root` user section.
- Deleting older database backup `# Delete older local database backup` section.
- Mount disk (optional section) under `# Mount disk` section.
- Stop application (optional section) under `# Stop tomcat` section.
- Clean logs (optional section) under `# Clean logs` section.
- Backup the database to the file system under `#Backup database` section.
- Backup repository and database dump to backup destination under `#Backup and purge old backups` section.
- Start application (optional section) `# Start tomcat` section.
- Show statistics (optional section) `# Status` section.
- Un mount disk (optional section) under `# Umount disk` section.

Configuration parameters explanation:

Parameter	Description
HOST	The server host name.
DATABASE_PASS	<p>The database password.</p> <div>  The data base user to do the backup is always root user. </div>
OPENKM_DB	<p>The application database name.</p> <div>  Usually will be named "okmdb". </div>
OPENKM_HOME	The openkm home folder.
TOMCAT_HOME	<p>The tomcat home folder.</p> <div>  Usually into OPENKM_HOME folder. </div>
DATABASE_EXP	<p>The database dump folder.</p> <div>  Usually into the same server. </div>
BACKUP_DIR	Backup destination.

Create a backup script **/root/backup.sh**

```
$ sudo su
$ vim /root/backup.sh
```

```
#!/bin/bash
#
## BEGIN CONFIG ##
HOST=$(uname -n)
DATABASE_PASS="*secret*"
```

```

OPENKM_DB="okmdb"
OPENKM_HOME="/home/openkm"
TOMCAT_HOME="$OPENKM_HOME/tomcat"
DATABASE_EXP="$OPENKM_HOME/db"
BACKUP_DIR="file:///mnt/backup"
## END CONFIG ##

# Check root user
if [ $(id -u) != 0 ]; then echo "You should run this script as root"; exit; fi

# Delete older local database backup
echo -e "### BEGIN: $(date +%x %X) ###\n"
rm -rf $DATABASE_EXP
mkdir -p $DATABASE_EXP

# Mount disk
if mount | grep "$BACKUP_DIR type" > /dev/null; then
    echo "$BACKUP_DIR already mounted";
else
    mount "$BACKUP_DIR";

    if mount | grep "$BACKUP_DIR type" > /dev/null; then
        echo "$BACKUP_DIR mounted";
    else
        echo "$BACKUP_DIR error mounting";
        exit -1;
    fi
fi

# Stop Tomcat
/etc/init.d/tomcat stop

# Clean logs
#echo "Clean Tomcat temporal files."
#rm -rf $TOMCAT_HOME/logs/*
#rm -rf $TOMCAT_HOME/temp/*
#rm -rf $TOMCAT_HOME/work/Catalina/localhost

# Backup database
if [ -n "$DATABASE_PASS" ]; then
    echo "* Backuping MySQL data from $OPENKM_DB..."
    mysqldump -h localhost -u root -p$DATABASE_PASS $OPENKM_DB > $DATABASE_EXP/mysql_$O
    echo "-----";
fi

# Backup and purge old backups
duplicity remove-older-than 1M --force $BACKUP_DIR/$HOST

if [ $(date +%u) -eq 7 ]; then
    echo "*** Full Backup ***"
    duplicity full --no-encryption $OPENKM_HOME $BACKUP_DIR/$HOST
    RETVAL=$?
else
    echo "*** Incremental Backup ***"
    duplicity --no-encryption $OPENKM_HOME $BACKUP_DIR/$HOST
    RETVAL=$?
fi

[ $RETVAL -eq 0 ] && echo "*** SUCCESS ***"
[ $RETVAL -ne 0 ] && echo "*** FAILURE ***"

# Start Tomcat
/etc/init.d/tomcat start
echo -e "\n### END: $(date +%x %X) ###"

# Status
echo "=====";

```

```

duplicity collection-status $BACKUP_DIR/$HOST
echo "*****";
df -h | grep "$BACKUP_DIR"
echo "===== ";

# Umount disk
sync
umount "$BACKUP_DIR"

```

Increase the backup period

Modify the **# Backup and purge old backups** section increasing the parameter value:



The "1M" value indicate will be removed backups older one month. Changing to "3M" or "1Y" indicate older 3 Month or 1 Year, etc..

```
--remove-older-than 3M
```

Perform the remote backup

Modify the **## Config section** section.

```

BACKUP_DIR="ftp://user@ftp.domain.es/backup"
export FTP_PASSWORD="WhateverPasswordYouSetUp"

```

Modify the **# Backup and purge old backups** section.

```

# Backup and purge old backups
duplicity remove-older-than 1M --force $BACKUP_DIR/$HOST

if [ $(date +%u) -eq 7 ]; then
    echo "*** Full Backup ***"
    duplicity full --no-encryption $OPENKM_HOME $BACKUP_DIR/$HOST
    RETVAL=$?
else
    echo "*** Incremental Backup ***"
    duplicity --no-encryption $OPENKM_HOME $BACKUP_DIR/$HOST
    RETVAL=$?
fi

[ $RETVAL -eq 0 ] && echo "*** SUCCESS ***"
[ $RETVAL -ne 0 ] && echo "*** FAILURE ***"

```

Modify the **# Status** section.

```

duplicity collection-status $BACKUP_DIR/$HOST
unset FTP_PASSWORD

```

Perform the remote backup to SMB or CIFS

Modify the **# Mount disk** section.

```
# Mount disk
if mount | grep "$BACKUP_DIR type" > /dev/null; then
    echo "$BACKUP_DIR already mounted";
else
    echo "Mounting $BACKUP_DIR ...";
    mount -t cifs //REMOTE_SERVER/REMOTE_DIR $BACKUP_DIR -o username=REMOTE_USER,password=REMOTE_PASS
fi

if mount | grep "$BACKUP_DIR type" > /dev/null; then
    echo "$BACKUP_DIR mounted";
else
    echo "Error mounting $BACKUP_DIR.";
    exit -1;
fi
fi
```

Do the PostgreSQL backup

Modify the **# Backup database** section

```
# Backup de PostgreSQL
echo "* Backeping PostgreSQL data from $OPENKM_DB..."
su postgres -c "pg_dump $OPENKM_DB" > $DATABASE_EXP/pg_$OPENKM_DB.sql

# Databases optimizations
su postgres -c "vacuumdb -a -z" > /dev/null
su postgres -c "reindexdb -a -q" 2> /dev/null
```

Do the remote backup to Amazon s3

Modify the **## Config** section.

```
BACKUP_DIR="s3+http://somebucket/somedirectory"
export AWS_ACCESS_KEY_ID=""
export AWS_SECRET_ACCESS_KEY=""
export PASSPHRASE=""
```

Modify the **# Backup and purge old backups** section.

```
# Backup and purge old backups
if [ $(date +%u) -eq 7 ]; then
    echo "*** Full Backup ***"
    duplicity full --no-encryption --s3-use-new-style $OPENKM_HOME $BACKUP_DIR/$HOST
    RETVAL=$?
else
    echo "*** Incremental Backup ***"
    duplicity --no-encryption --s3-use-new-style $OPENKM_HOME $BACKUP_DIR/$HOST
    RETVAL=$?
fi

[ $RETVAL -eq 0 ] && echo "*** SUCCESS ***"
```

Modify the **# Status** section.

```
# Status
```



```

duplicity collection-status --s3-use-new-style $BACKUP_DIR/$HOST
unset AWS_ACCESS_KEY_ID
unset AWS_SECRET_ACCESS_KEY
unset PASSPHRASE

```

Configure crontab

To install the cron job, run:

```

$ sudo mkdir /root/logs
$ sudo crontab -e

```

Add these lines according to your personal configuration:

```

MAILTO=nomail@openkm.com
@daily /root/backup.sh | tee /root/logs/backup.$(date +%Y.%m.%d_%H.%M.%S).log

```



More information at [Crontab quick reference](#)

If you want to be notified by mail you should install "**postfix**" service in your server.

Troubleshooting deleting old backups

You have configured Duplicity to make a full backup every 7 days, and deleting backups older than 14 days to save space. What happen if you see a message like this?

```

Tue May 1 00:00:00 2012
Wed May 2 00:00:00 2012
Which can't be deleted because newer sets depend on them

```

Duplicity uses rsync, which contains incremental changes. Those files won't be deleted because, even though the backup maybe older than 7 days, there are backups which are incremental and younger than 7 days.

So, after 2 weeks have passed, those files will be deleted, since the full backup and the incremental backups are now 14 days old, and there exists a full backup newer than the full and incremental backups.

Restoring backup

You can list the available backups:

```

$ duplicity collection-status /path/to/backup

Sun Sep 11 05:00:19 2012    4.25 GB    4.25 GB (current mirror)
Sun Sep 4 00:00:18 2012   13.3 MB    4.26 GB
Sun Aug 28 00:00:13 2012   674 MB    4.92 GB
Sun Aug 21 00:00:14 2012   5.50 MB    4.93 GB

```

Sun Aug 14 00:00:16 2012	1.75 MB	4.93 GB
Sun Aug 7 00:00:12 2012	288 KB	4.93 GB
Sun Jul 31 00:00:13 2012	43.0 KB	4.93 GB
Fri Jul 29 10:36:39 2012	5.56 KB	4.93 GB

Then decide to restore some of the backups, for example the one made on Sun Aug 28 00:00:13 2012:

```
$ duplicity restore --restore-time 2012-08-28 /path/to/backup /path/to/destination
```



The `--restore-time` parameter accepts several formats. See [duplicity](#) documentation and [Duplicity Backup Howto](#) for more info.



Inside of `/path/to/destination` you should see a directory `/home/openkm`, and inside it a couple of directories:

- **db**: The backup of the database.
- **tomcat**: The backup of the Tomcat installation and OpenKM repository into.

Additional information

- [Using Duplicity to Backup Your Data](#)
- [Ubuntu/Debian - Encrypted Incremental Backups With Duplicity on Amazon S3](#)
- [Duplicity Backup Howto](#)
- [Duplicity URL Format](#)
- [Installation and configuration of duplicity for encrypted SFTP remote backup](#)
- [Playing with Duplicity backup and restore tool and Amazon S3](#)
- [Bash wrapper script for automated backups with duplicity](#)

Linux backup and restore with rsync

Rsync is a software application and network protocol for Unix, Linux and Windows systems which synchronizes files and directories from one location to another while minimizing data transfer using delta encoding when appropriate. An important feature of rsync not found in similar programs/protocols is that the mirroring takes place with only one transmission in each direction. rsync can copy or display directory contents and copy files, optionally using compression and recursion.

Preliminaries

In these scripts are assumed these default values:

- **\$TOMCAT_HOME** value is `"/home/openkm/tomcat"`
- OpenKM **database** is named `"okmdb"`.
- OpenKM **database password** value is `"*secret*"`.
- OpenKM start & stop **service script** location is `"/etc/init.d/tomcat"`

Sample script



As a good practice the backup should be done by root user.



The script below does a backup to USB and the application database is MySQL.


USB disk mount point can be defined in `/etc/fstab` as:

```
/dev/sdb1    /mnt/backup    ext4    defaults    0    0
```

These are the global script sections:

- Global var configuration under **## BEGIN CONFIG ##** section.
- Checking user root under **# Check root** user section.
- Deleting older database dump **# Delete older local database backup** section.
- Mount disk (optional section) under **# Mount disk** section.
- Stop application (optional section) under **# Stop tomcat** section.
- Clean logs (optional section) under **# Clean logs** section.
- Backup the database to the filesystem under **#Backup database** section.
- Backup repository and database dump to backup destination under **# Create backup** section.
- Start application (optional section) **# Start tomcat** section.
- Umount disk (optional section) under **# Umount disk** section.

Configuration parameters explanation:

Parameter	Description
HOST	The server host name.
DATABASE_PASS	<p>The database password.</p> <div>  The database user to do backup is always root user. </div>
OPENKM_DB	<p>The application database name.</p> <div>  Usually will be named "okmdb". </div>
OPENKM_HOME	The openkm home folder.
TOMCAT_HOME	<p>The tomcat home folder.</p> <div>  Usually into OPENKM_HOME folder. </div>
DATABASE_EXP	<p>The database dump folder.</p> <div>  Usually into the same server. </div>
BACKUP_DIR	Backup destination.
RSYNC_OPTS	Rsync parameters.

Create a backup script **/root/backup.sh**

```
$ sudo su
$ vim /root/backup.sh
```

```
#!/bin/bash
```

```

#
## BEGIN CONFIG ##
HOST=$(uname -n)
DATABASE_PASS="*secret*"
OPENKM_DB="okmdb"
OPENKM_HOME="/home/openkm"
TOMCAT_HOME="$OPENKM_HOME/tomcat"
DATABASE_EXP="$OPENKM_HOME/db"
BACKUP_DIR="/mnt/backup"
RSYNC_OPTS="-apzhR --stats --delete --exclude=*~ --delete-excluded"
## END CONFIG ##

# Check root user
if [ $(id -u) != 0 ]; then echo "You should run this script as root"; exit; fi

# Delete older local database backup
echo -e "### BEGIN: $(date +"%x %X") ###\n"
rm -rf $DATABASE_EXP
mkdir -p $DATABASE_EXP

# Mount disk
if mount | grep "$BACKUP_DIR type" > /dev/null; then
    echo "$BACKUP_DIR already mounted";
else
    mount "$BACKUP_DIR";

    if mount | grep "$BACKUP_DIR type" > /dev/null; then
        echo "$BACKUP_DIR mounted";
    else
        echo "$BACKUP_DIR error mounting";
        exit -1;
    fi
fi

# Stop Tomcat
/etc/init.d/tomcat stop

# Clean logs
#echo "Clean Tomcat temporal files."
#rm -rf $TOMCAT_HOME/logs/*
#rm -rf $TOMCAT_HOME/temp/*
#rm -rf $TOMCAT_HOME/work/Catalina/localhost

# Backup database
if [ -n "$DATABASE_PASS" ]; then
    echo "* Backeping MySQL data from $OPENKM_DB..."
    mysqldump -h localhost -u root -p$DATABASE_PASS $OPENKM_DB > $DATABASE_EXP/mysql_$O
    echo "-----";
fi

# Create backup
rsync $RSYNC_OPTS $OPENKM_HOME $BACKUP_DIR/$HOST

# Start Tomcat
/etc/init.d/tomcat start
echo -e "\n### END: $(date +"%x %X") ###"

# Umount disk
sync
umount "$BACKUP_DIR"

```

Do backup with rotation

Modify the **# Create backup** section.

```
# Create backup
# Calculate snapshot
LAST_SNAPSHOT=`ls -ltr $BACKUP_DIR | tail -1 | awk {'print $8'} | cut -d . -f 2`
NEW_SNAPSHOT=$((LAST_SNAPSHOT+1))

rsync $RSYNC_OPTS --link-dest="$BACKUP_DIR/$HOST/backup.$LAST_SNAPSHOT" $FILES "$BACKUP_DIR/$HOST/backup.$NEW_SNAPSHOT"
```



Other configuration:

```
# Create backup
"cd $BACKUP_DIR/$HOST; rm -rf backup.3; mv backup.2 backup.3; mv backup.1 backup.0;
rsync $RSYNC_OPTS --link-dest="$BACKUP_DIR/$HOST/backup.1" "$OPENKM_HOME" "$BACKUP_DIR/$HOST/backup.0"
```

Do remote backup

Modify the **## BEGIN CONFIG ##** section.

```
BACKUP_SRV="user@server.com"
```

Modify the **# Create backup** section.

```
# Create backup
# Create missing backup folder ( optional )
ssh $BACKUP_SRV "mkdir -p $BACKUP_DIR"

# Calculate snapshot
NEW_SNAPSHOT=$(date "+%Y-%m-%dT%H:%M:%S")
LAST_SNAPSHOT=$(ssh $BACKUP_SRV ls -ltr $BACKUP_DIR/$HOST | tail -1)

if [ $LAST_SNAPSHOT ]; then
    echo "Incremental backup $NEW_SNAPSHOT based on $LAST_SNAPSHOT... "
    rsync $RSYNC_OPTS --link-dest="$BACKUP_DIR/$HOST/$LAST_SNAPSHOT" "$OPENKM_HOME" "$BACKUP_DIR/$HOST/$NEW_SNAPSHOT"
else
    echo "Initial full backup $NEW_SNAPSHOT..."
    rsync $RSYNC_OPTS "$OPENKM_HOME" "$BACKUP_SRV:$BACKUP_DIR/$HOST/$NEW_SNAPSHOT"
fi
```



More simple configuration:

```
rsync $RSYNC_OPTS "$OPENKM_HOME" "$BACKUP_SRV:$BACKUP_DIR/$HOST"
```

Do incremental backup with backup-dir

Modify the **## BEGIN CONFIG ##** section.

```
RSYNC_OPTS="-ahR --partial --stats --delete --exclude=*~ --exclude=tomcat/temp/* --exclude=*.log"
```

Modify the **# Create backup** section.

```
# Create backup
NEW_SNAPSHOT=`date +%Y-%m-%d`
echo -e "\n### RSYNC - BEGIN : $(date +%x %X) ###\n"
echo "* New Snapshot: $NEW_SNAPSHOT"
rsync $RSYNC_OPTS --backup --backup-dir="$BACKUP_DIR/$HOST/INCREMENTAL/$NEW_SNAPSHOT"
```

Do incremental backup with link-dest

Modify the **## BEGIN CONFIG ##** section.

```
RSYNC_OPTS="--ahR --partial --stats --delete --exclude=*~ --exclude=tomcat/temp/* --ex
```

Modify the **# Create backup** section.

```
# Create backup
# Calculate snapshot
LAST_SNAPSHOT=`ssh $BACKUP_SRV "cd $BACKUP_DIR && stat --format='%n' * | tail -1"`
NEW_SNAPSHOT=`date +%Y-%m-%d`
echo -e "\n### RSYNC - BEGIN : $(date +%x %X) ###\n"
echo "* Last Snapshot: $LAST_SNAPSHOT"
echo "* New Snapshot: $NEW_SNAPSHOT"

if [ ! $LAST_SNAPSHOT ]; then
    rsync $RSYNC_OPTS $OPENKM_HOME/ "$BACKUP_DIR/$HOST/$NEW_SNAPSHOT"
else
    rsync $RSYNC_OPTS --link-dest="$BACKUP_DIR/$HOST/$LAST_SNAPSHOT" $OPENKM_HOME/ "$BA
fi
```

Do remote backup to SMB or CIFS

Modify the **# Mount disk** section.

```
# Mount disk
if mount | grep "$BACKUP_DIR type" > /dev/null; then
    echo "$BACKUP_DIR already mounted";
else
    echo "Mounting $BACKUP_DIR ...";
    mount -t cifs //REMOTE_SERVER/REMOTE_DIR $BACKUP_DIR -o username=REMOTE_USER,password=
    if mount | grep "$BACKUP_DIR type" > /dev/null; then
        echo "$BACKUP_DIR mounted";
    else
        echo "Error mounting $BACKUP_DIR.";
        exit -1;
    fi
fi
```

Do PostgreSQL backup

Modify the **# Backup database** section

```
# Backup de PostgreSQL
echo "* Backeping PostgreSQL data from $OPENKM_DB..."
```

```
su postgres -c "pg_dump $OPENKM_DB" > $DATABASE_EXP/pg_$OPENKM_DB.sql

# Databases optimizations
su postgres -c "vacuumdb -a -z" > /dev/null
su postgres -c "reindexdb -a -q" 2> /dev/null
```

Configure crontab

To install the cron job, run:

```
$ sudo mkdir /root/logs
$ sudo crontab -e
```

And add these lines according to your personal configuration:

```
MAILTO=nomail@openkm.com
@daily /root/backup.sh | tee /root/logs/backup.$(date +%Y.%m.%d_%H.%M.%S).log
```



More information at [Crontab quick reference](#)

If you want to be notified by mail you should install "**postfix**" service in your server.

Rsync and FTP

Rsync need SSH access to work, but you can also use it with an FTP server if you mount it locally with **curlftps**:



Backup performance may degrade using this method.

Install curlftps and rsync:

```
$ sudo apt-get install curlftps rsync
```

Create mountpoin:

```
$ mkdir /path/to/mount
```

Mount remote director:

```
$ curlftps ftp.example.com /path/to/mount
```

Create local directory:

```
$ mkdir /local/path
```

Rsync mounted directory to local director:

```
$ rsync -r -t -v --progress --bwlimit=500 /path/to/mount/ /local/path/
```


Restoring backup

You can list the available backups:

```
$ ls -ld /path/to/backup/*  
  
drwxr-xr-x 4 root root 4096 sep 11 05:00:19 2012 2012-09-11T05:00:19  
drwxr-xr-x 4 root root 4096 sep  4 00:00:18 2012 2012-09-04T00:00:18  
drwxr-xr-x 4 root root 4096 aug 28 00:00:13 2012 2012-08-28T00:00:13  
drwxr-xr-x 4 root root 4096 aug 21 00:00:14 2012 2012-08-21T00:00:14  
drwxr-xr-x 4 root root 4096 aug 14 00:00:16 2012 2012-08-14T00:00:16  
drwxr-xr-x 4 root root 4096 aug  7 00:00:12 2012 2012-08-07T00:00:12  
drwxr-xr-x 4 root root 4096 jul 31 00:00:13 2012 2012-07-31T00:00:13  
drwxr-xr-x 4 root root 4096 jul 29 10:36:39 2012 2012-07-29T10:36:39
```

Then decide to restore some of the backups, for example the one made on Sun Aug 28 00:00:13 2012:

```
$ cp /path/to/backup/2012-08-28T00:00:13 /path/to/destination
```



See [rsync](#) documentation for more info.



Inside of **/path/to/destination** you should see a directory **/home/openkm**, and inside it a couple of directories:

- **db**: The backup of the database.
- **tomcat**: The backup of the Tomcat installation and OpenKM repository into.

Additional information

- [Easy Automated Snapshot-Style Backups with Linux and Rsync](#)

Linux backup with LFTP

LFTP is a sophisticated FTP client, and a file transfer program supporting a number of network protocols.



This configuration does not provide incremental backups.

Preliminaries

In these scripts are assumed these default values:

- **\$TOMCAT_HOME** value is `"/home/openkm/tomcat"`
- OpenKM **database** is named `"okmdb"`.
- OpenKM **database password** value is `"*secret*"`.
- OpenKM start & stop **service script** location is `"/etc/init.d/tomcat"`

Sample script



As a good practice the backup should be done by root user.



The script below does a backup to USB and the application database is MySQL.

USB disk mount point can be defined in `/etc/fstab` as:





```
/dev/sdb1    /mnt/backup    ext4    defaults    0    0
```

These are the global script sections:

- Global var configuration under **## BEGIN CONFIG ##** section.
- Checking user root under **# Check root** user section.
- Deleting older database backup **# Delete older local database backup** section.
- Stop application (optional section) under **# Stop tomcat** section.
- Clean logs (optional section) under **# Clean logs** section.
- Backup the database to the filesystem under **#Backup database** section.
- Backup repository and database dump to backup destination under **#Backup and purge old backups** section.
- Start application (optional section) **# Start tomcat** section.

Configuration parameters explanation:

Parameter	Description
-----------	-------------

HOST	The server host name.
NOW	Actual date in format 'dd-MM-yyyy'.
DATABASE_PASS	<p>The database password.</p> <div>  The data base user to do backup is always root user. </div>
OPENKM_DB	<p>The application database name.</p> <div>  Usually will be named "okmdb". </div>
OPENKM_HOME	The openkm home folder.
TOMCAT_HOME	<p>The tomcat home folder.</p> <div>  Usually into OPENKM_HOME folder. </div>
DATABASE_EXP	<p>The database dump folder.</p> <div>  Usually into the same server. </div>
BACKUP_DIR	FTP Backup destination.

Create a backup script **/root/backup.sh**

```
sudo su
vim /root/backup.sh
```

```
#!/bin/bash
#
## BEGIN CONFIG ##
HOST=$(uname -n)
NOW=$(date +"%d-%m-%Y")
```

```

DATABASE_PASS="*secret*"
OPENKM_DB="okmdb"
OPENKM_HOME="/home/openkm"
TOMCAT_HOME="$OPENKM_HOME/tomcat"
DATABASE_EXP="$OPENKM_HOME/db"
BACKUP_DIR="ftp://user:password@ftp.server/backup"
## END CONFIG ##

# Check root user
if [ $(id -u) != 0 ]; then echo "You should run this script as root"; exit; fi

# Delete older local database backup
echo -e "### BEGIN: $(date +%x %X)" ###\n"
rm -rf $DATABASE_EXP
mkdir -p $DATABASE_EXP

# Stop Tomcat
/etc/init.d/tomcat stop

# Clean logs
#echo "Clean Tomcat temporal files."
#rm -rf $TOMCAT_HOME/logs/*
#rm -rf $TOMCAT_HOME/temp/*
#rm -rf $TOMCAT_HOME/work/Catalina/localhost

# Backup database
if [ -n "$DATABASE_PASS" ]; then
    echo "* Backuping MySQL data from $OPENKM_DB..."
    mysqldump -h localhost -u root -p$DATABASE_PASS $OPENKM_DB > $DATABASE_EXP/mysql_$O
    echo "-----";
fi

# Backup and purge old backups
lftp $BACKUP_DIR -e "mkdir $NOW; mirror --reverse $OPENKM_HOME $NOW; exit"

# Start Tomcat
/etc/init.d/tomcat start
echo -e "\n### END: $(date +%x %X)" ###

```

Configure crontab

To install the cron job, run:

```

$ sudo mkdir /root/logs

$ sudo crontab -e

```

Add these lines according to your personal configuration:

```

MAILTO=nomail@openkm.com
@daily /root/backup.sh | tee /root/logs/backup.$(date +%Y.%m.%d_%H.%M.%S).log

```

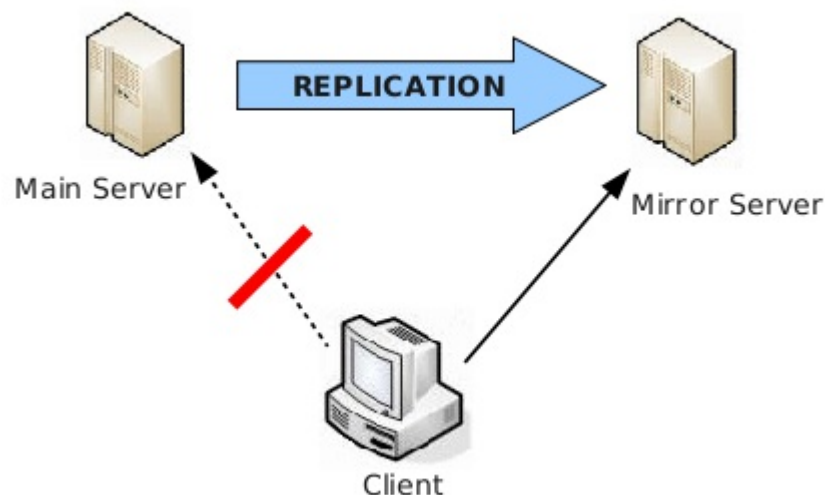


More information at [Crontab quick reference](#)

If you want to be notified by mail you should install "**postfix**" service in your server.

Instance replication

To enhance OpenKM availability you can have two instances of the application running on different servers. If the principal server gets down due to a hardware failure you can switch to the mirrored server and keep working.



In this scheme, we assume:

- Each node has its own document repository and database.
- The backup instance is supposed to be a clone of the main instance.
- In case the main instance is down, you should send the client request to the backup instance.



An option is to use a DNS which you can modify in case of server failure, or an HAProxy instance which properly configured can detect a problem in the main server and forward the request to the working one.



Keep in mind that in this configuration only one server should accept client petitions.

When the main server is down, the backup one becomes the new main server so the old main server is now the backup one (well, when it is available again).

This is the configuration of each server:

Main server

- OpenKM is configured normally.
- The database is configured to send modifications to the backup instance.
- Rsync is configured to send datastore modifications every few minutes to the backup server.

Mirror server

- OpenKM is configured in read-only mode.
- The database is configured to accept modification from the main instance.
- Datastore files are periodically updated by Rsync tool from the main server.

Different implementations

The provided solution is not the only one, but the simplest.

An improvement over this first approach is using an external file server (connected by NFS) so Rsync is not needed. Using a NAS solution which improves the data backup and availability. It is highly recommended if the stored information is critical for your business.

Another improvement is related to the database. Some database can be configured in master / slave mode, which the required mode for this server configuration. But you can also deploy the database in a dedicated server (which should be also replicated) but some providers like Oracle offer specific database cluster solutions which should be taken in consideration.

Sample Linux implementation



The replication is done only in one direction, from main server to mirror server.

The mirror server must be configured to be in read-only mode so users can't add or modify documents.

The follow script will propagate the repository changes from main server to mirror server:



On the sample the main server 192.168.1.101.

Both OpenKM are installed at the OpenKM user home (/home/openkm) for simplicity.

This script should be executed - by user root - from the main server, and can be scheduled to be executed every day using Linux cron utility, for example.

The script uses rsync to minimize network load and only modified or added documents will be transferred. The whole process can take a few minutes, depending on your repository activity.

```
#!/bin/bash
echo -e "### BEGIN: $(date +"%x %X") ###\n"

# Stop local Tomcat
/etc/inid.d/tomcat stop

# Stop remote Tomcat
ssh root@192.168.1.101 '/etc/inid.d/tomcat stop'

# Sync OpenKM repositories
rsync -rahe 'ssh -p 22022' --stats --delete --exclude tomcat/server/default/log --exc

# Start local Tomcat
/etc/init.d/tomcat start

# Start remote Tomcat
```

```
ssh root@192.168.1.101 '/etc/init.d/tomcat start'  
echo -e "\n### END: $(date +"%x %X") ###"
```

Copy the following SQL into the remote /home/tomcat-7.0.61/start.sql server location.

```
update OKM_CONFIG set 'true' where CFG_KEY='system.readonly';
```



After the first synchronization, stop remote Tomcat, update the start.sql script there and start the Tomcat again.

Setup



This section provides information for **configuring** and **tuning** OpenKM.

The application contains a lot of [Configuration parameters](#) and probably you will not need to know all of them. However, we encourage you to take a look at the [Recommended configuration parameters](#) section where you will find the minimum configuration required to use OpenKM.

Also you could be interested on:

- [Preview parameters](#)
- [User interface configuration parameters](#)

OpenKM works with third-party applications Wordpress, Joomla or Google Drive among others that need specific setup to get it running.



If you have some issue see [Troubleshooting](#) section.

Antivirus configuration

OpenKM can be integrated with an anti virus software which is be able to execute anti virus check from command line.

The example below is based on [ClamAV](#), an Open Source anti virus engine which detects trojans, viruses, malware & other malicious programs.

Linux

Debian / Ubuntu

Install antivirus:

```
$ sudo aptitude install clamav
```

Go to **Administration > Configuration parameters:**

Field / Property	Type	Description
system.antivir	String	/usr/bin/clamscan

Red Hat / CentOS

ClamAV is not present by default in default repositories, so you must install it from a non default one.

Repository valid for Centos 5.2 (for other versions it may be different):

```
[dag]
name=Dag RPM Repository for Red Hat Enterprise Linux
baseurl=http://apt.sw.be/redhat/el$releasever/en/$basearch/dag/
gpgcheck=1
gpgkey=http://dag.wieers.com/packages/RPM-GPG-KEY.dag.txt
enabled=1
```

Install antivirus:

```
$ yum install clamd
```

Start the daemon:

```
$ /etc/init.d/clamd start
```

Update the virus database:

```
$ freshclam
```

Windows

To Install and configure ClamAV on Windows:

Note: **GREEN Text** like this denotes commands run in a command console/terminal

BLACK Text like this denotes code changes

- Download the latest version from: <https://www.clamav.net/downloads>
- Unpack the ZIP archive and move it to: C:\ so that it appears in your path as: C:\ClamAV
- Copy the example conf files from C:\ClamAV\conf_examples to C:\ClamAV
- Open up a command terminal: Start->Run **cmd**
- Change to the ClamAV directory: **cd C:\ClamAV**
- Run **clamconf.exe**, it should inform that you need to edit the example config file. (I included this just so you can check that the files are in the right location)

For DEFAULT MINIMAL configuration (Probably not suitable and you should configure the parameters for your situation)

Edit **freshclam.conf**, **clamav-milter.conf** and comment **Example**. In addition to commenting **Example** in **clamd.conf**; to enable the local server you will need to change line 80 to: **LocalSocket C:\CLAMAV\clamd.socket**

- Run **clamconf.exe** again, you will get an output of all your configurations.



At the beginning you will get a message about being unable to open database, this is because it has not yet been created.

- Run **mkdir database**. If you run **clamconf** again you will see that the database error is gone and that you have 0 signatures
- Run **freshclam.exe**, this will start the initial signature download for ClamAV and may take a few minutes.

At this point you can run **clamd.exe** to check that the server can run manually. (If you decide to test run at this point, you will have to kill the process for the next section)

In case you need to create a **freshclam.conf** file use this as base:

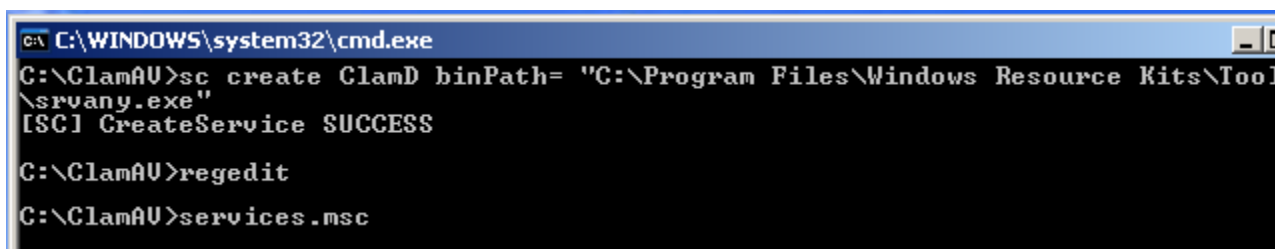
```
# URL of server where database updates are to be downloaded from
# If this option is given multiple times, each will be tried in
```

```
# the order given until an update is successfully downloaded
#MAKE SYU
DatabaseDirectory "C:\Program Files\ClamAV-x64\db\"
DatabaseMirror database.clamav.net
# Number of times to try each mirror before moving to the next one
MaxAttempts 3
#LogFile C:\Program Files\ClamAV-x64\log\freshclam.log
#NotifyClamd C:\Program Files\ClamAV-x64\clamd.conf
LogFileMaxSize 20480000
LogTime true
UpdateLogFile C:\Program Files\ClamAV-x64\log\freshclam.log
```

Clamd Service Setup

Now to setup it up so it starts automatically:

Details for the screen capture are below:



```
C:\WINDOWS\system32\cmd.exe
C:\ClamAU>sc create ClamD binPath= "C:\Program Files\Windows Resource Kits\Tools\
\srwany.exe"
[SC] CreateService SUCCESS
C:\ClamAU>regedit
C:\ClamAU>services.msc
```

- Run **sc create ClamD binPath= C:\Program Files\Windows Resource Kits\Tools\srwany.exe** (yes the space is required in this command and you may have your resource tools in a different location)

It should tell you: [SC] CreateService SUCCESS

- Run **Regedit**. Find: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ClamD
 - Right click on ClamD and create a new Key
 - Name the key **Parameters**
 - Click on the new Parameters Key, in the right pane create a new String Value
 - Name the String Value **Application**
 - Right click and Modify Application to: **C:\ClamAV\clamd.exe -c C:\ClamAV\clamd.conf**
 - Close Regedit
- Run **services.msc**
 - Find ClamD in the list
 - Right click -> Properties
 - Change Startup type to: Automatic
 - Select the recovery tab and change all three failures to: Restart the Service
 - Select the General tab and Apply

Freshclam Virus Signature Auto Update

Almost done!

- *Navigate to: Start->All Programs->Accessories->System Tools->Scheduled Tasks*
- *Add Scheduled Task*
- *Next->Browse to C:\ClamAV\freshclam.exe*
- *Select to Perform this task daily*
- *Next*
- *Select a good time to update*
- *Next*
- *Enter the user credentials to run as*
- *Next*
- *Select open Advance options*
- *Next*
- *Alter Run to be: **C:\ClamAV\freshclam.exe -c C:\ClamAV\freshclam.conf***
- *Select the Settings Tab*
- *Change the Stop Task to 30 minutes*
- *Apply*
- *Ok*

Right click and run the freshclam task to make sure you do not have any errors.

*Go to **Administration > Configuration parameters:***

<i>Field / Property</i>	<i>Type</i>	<i>Description</i>
system.antivir	String	C:\ClamAV\clamscan.exe

If everything was configured correctly, OpenKM will scan every uploaded document in order to check if it's safe.

Cleaning older activity log data

There's a table named `OKM_ACTIVITY` where openkm saves a log with the actions done by the users on all the nodes. In this table can be easily several millions of rows what can affect the performance of the database. Really you should be interested on most actual activity log data - for example last year - and keep the other data outside the database. For it is a good practice to perform some cleaning process, exporting oldest activity log data and then deleting it from database.



Taking control of `OKM_ACTIVITY` log table size you can improve your database performance and reduce the total amount of your database backup size and time for doing it.

Doing backups of `OKM_ACTIVITY` by range and deleting from database you keep your activity log information safe without affecting database performance.

You always can restore the `OKM_ACTIVITY` logs in other database for doing forensic analytics.

MySQL

Linux

Create a file named **`clean_okm_activity.sh`** at `/root/` folder:

```
#!/bin/sh
## BEGIN CONFIG ##
DATABASE_PASS="*secret*"
BACKUP_DIR="/mnt/backup/okm_activity"
## END CONFIG ##

mysqldump -h localhost -u root -p$DATABASE_PASS -w"ACT_DATE < (NOW() - INTERVAL 365 D.
mysql -h localhost -u root -p$DATABASE_PASS okmdb < /root/delete.sql
```

Make **`clean_okm_activity.sh`** as executable file:

```
$chmod +x clean_okm_activity.sh
```

Create a file named **`delete.sql`** at `/root/` folder:

```
DELETE FROM OKM_ACTIVITY WHERE ACT_DATE < (NOW() - INTERVAL 365 DAY);
ANALYZE TABLE OKM_ACTIVITY;
```

Configure crontab

To install the cron job, run:

```
$ sudo mkdir /root/logs
$ sudo crontab -e
```

And add these lines according to your personal configuration:

```
MAILTO=nomail@openkm.com
@daily /root/clean_okm_activity.sh | tee /root/logs/okm_activity.$(date +%Y.%m.%d)
```



More information at [Crontab quick reference](#)

If you want to be notified by mail you should install "**postfix**" service in your server.

Windows

Create a file named **clean_okm_activity.bat** at c:\backup folder:

```
set SQLFILE_DATE=%DATE:~6,4%.%DATE:~3,2%.%DATE:~0,2%
set SQLFILE_TIME=%TIME:~0,2%.%TIME:~3,2%
set SQLFILE=c:\backup\okm_activity-%SQLFILE_DATE%-%SQLFILE_TIME%.sql
set DATABASE_PASS=*secret*

mysqldump -h localhost -u root -p%DATABASE_PASS% -w"ACT_DATE < (NOW() - INTERVAL 365
mysql -h localhost -u root -p%DATABASE_PASS% okmdb < delete.sql
```

Create a file named **delete.sql** at c:\backup folder:

```
DELETE FROM OKM_ACTIVITY WHERE ACT_DATE < (NOW() - INTERVAL 365 DAY);
ANALYZE TABLE OKM_ACTIVITY;
```



You can use Windows Task Scheduler for configuring periodically your task.

Database replication with federated tables

The procedure for using *FEDERATED* tables is very simple. Normally, you have two servers running, either both on the same host or on different hosts. (It is possible for a *FEDERATED* table to use another table that is managed by the same server, although there is little point in doing so.)



You should have two database servers. Production MySQL server (A) and backup MySQL server (B).

First of all must enable federated tables on production server. Stop MySQL production service and edit the MySQL configuration file (*my.cnf* or *my.ini* depending on your OS). Go to *[mysqld]* section and set below:

```
[mysqld]
# To enable federated feature
federated
```

Start your production MySQL service and execute from your preferred MySQL client tool the command:

```
show create table okm_activity;
```

You will get something like:

```
CREATE TABLE `okm_activity` (
  `ACT_ID` bigint(20) NOT NULL AUTO_INCREMENT,
  `ACT_ACTION` varchar(127) COLLATE utf8_bin DEFAULT NULL,
  `ACT_DATE` datetime DEFAULT NULL,
  `ACT_ITEM` varchar(64) COLLATE utf8_bin DEFAULT NULL,
  `ACT_PARAMS` varchar(4000) COLLATE utf8_bin DEFAULT NULL,
  `ACT_PATH` longtext COLLATE utf8_bin,
  `ACT_TENANT` bigint(20) DEFAULT NULL,
  `ACT_USER` varchar(64) COLLATE utf8_bin DEFAULT NULL,
  PRIMARY KEY (`ACT_ID`),
  KEY `IDX_ACTIVITY_TENANT` (`ACT_TENANT`),
  KEY `IDX_ACTIVITY_DATE` (`ACT_DATE`),
  KEY `IDX_ACTIVITY_TNTUSRACT` (`ACT_TENANT`,`ACT_USER`,`ACT_ACTION`),
  KEY `IDX_ACTIVITY_TNTDATACT` (`ACT_TENANT`,`ACT_DATE`,`ACT_ACTION`)
) ENGINE=innodb DEFAULT CHARSET=utf8 COLLATE=utf8_bin
```

Create the table "**okm_activity**" in your backup MySQL server.

Create the federated table in your production MySQL server (take a look we have **changed the table name**, the **ENGINE=innodb** for **ENGINE=federated** and we have **added the connection link**):

```
CREATE TABLE `okm_activity_remote` (
  `ACT_ID` bigint(20) NOT NULL AUTO_INCREMENT,
  `ACT_ACTION` varchar(127) COLLATE utf8_bin DEFAULT NULL,
  `ACT_DATE` datetime DEFAULT NULL,
  `ACT_ITEM` varchar(64) COLLATE utf8_bin DEFAULT NULL,
  `ACT_PARAMS` varchar(4000) COLLATE utf8_bin DEFAULT NULL,
  `ACT_PATH` longtext COLLATE utf8_bin,
  `ACT_TENANT` bigint(20) DEFAULT NULL,
  `ACT_USER` varchar(64) COLLATE utf8_bin DEFAULT NULL,
  PRIMARY KEY (`ACT_ID`),
  KEY `IDX_ACTIVITY_TENANT` (`ACT_TENANT`),
  KEY `IDX_ACTIVITY_DATE` (`ACT_DATE`),
  KEY `IDX_ACTIVITY_TNTUSRACT` (`ACT_TENANT`,`ACT_USER`,`ACT_ACTION`),
  KEY `IDX_ACTIVITY_TNTDATACT` (`ACT_TENANT`,`ACT_DATE`,`ACT_ACTION`)
) ENGINE=FEDERATED DEFAULT CHARSET=utf8 COLLATE=utf8_bin
CONNECTION='mysql://user:password@server_b:3306/okmdb/okm_activity';
```

Finally you can periodically execute an script like this one from your MySQL production server:

```
select (NOW() - INTERVAL 1 DAY) into @dump_date;
insert into okm_activity_remote select * from okm_activity where act_date < @dump_date;
delete from okm_activity where act_date < @dump_date;
analyze table okm_activity;
```

Configuring Apache HTTP Reverse-Proxy

Exposing OpenKM directly from Tomcat can be risky, if you need the application to be accessed from the Internet (for example <https://issues.jboss.org/browse/JBAS-3861>). It could be infected by [PerlBot](#). 8080 may be blocked by a firewall, for this reason is a recommendable to expose your OpenKM installation through the standard web port 80. In the following steps we explain how to configure Apache to handle these requests and forward to Tomcat application server using the AJP13 protocol.

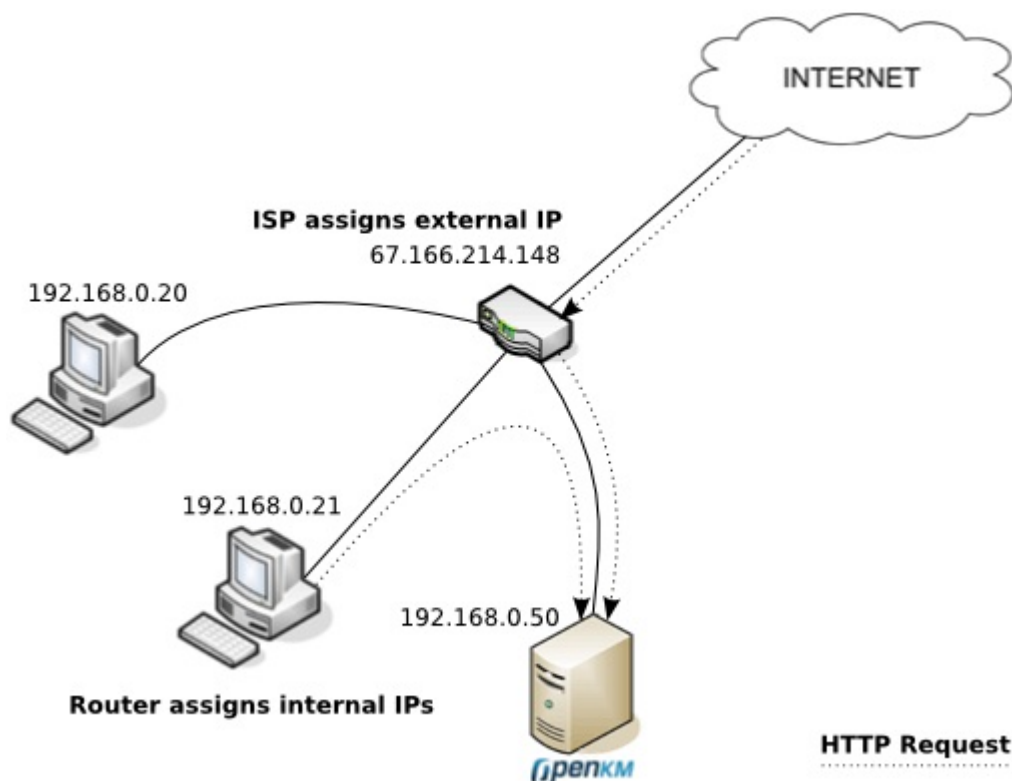
From the Apache documentation: The AJP13 protocol is packet-oriented. A binary format was presumably chosen over the more readable plain text for performance reasons. The web server communicates with the servlet container over TCP connections. To cut down on the expensive process of socket creation, the web server will attempt to maintain persistent TCP connections to the servlet container, and to reuse a connection for multiple request/response cycles.

Internal IP vs external IP

Your OpenKM can be accessed from two different zones: Internet and LAN. This means that to access this server you need to use two IPs: external IP (Internet) and internal IP (LAN).

The internal IP address (also known as "local IP address") is the address that is assigned by your local network router that often begins with 192.168.x.x. These IP addresses can only be seen by other computers in your local network (LAN) and not by any computers connected in an external network such the Internet.

To reach the Internet or a computer in another network your computer is often assigned an external IP address, which can then be used to refer to the computer in your local network.



In the above picture, there are three computers in the local network that have each been assigned their own internal IP address by the router. The ISP is connected to the router and gives the router an external IP address that allows it to communicate with the Internet. On the Internet everyone sees your external IP address, but any information coming from the router is "converted" from the external IP address to the internal IP address.

So if you want your OpenKM installation accessible from both LAN and Internet, the trick here is to configure the client computers to resolve your **internal IP** (192.168.0.50) if they are inside the LAN or the **external IP** if they are on Internet. To resolve the Internet IP (67.166.214.148) your computer uses the public DNS. So, you need to configure a sort of DNS server inside the LAN or modify every client [host file](#) to resolve to the internal IP.

Debian and Ubuntu

The first thing is to install the required Apache software. From Debian / Ubuntu you can install Apache with a single command:

```
$ sudo apt-get install apache2
```

Edit the file named `/etc/apache2/apache2.conf` and configure a `ServerName` to prevent warnings in the Apache startup process:

```
$ vim /etc/apache2/apache2.conf
```

```
ServerRoot "/etc/apache2"
ServerName "your-domain.com"
```



Edit apache2.conf is optional.

Enable the proxy module:

```
$ sudo a2enmod proxy_ajp
```

Create the configuration file `/etc/apache2/sites-available/openkm.conf` with this content:

```
$ vim /etc/apache2/sites-available/openkm.conf
```

```
<VirtualHost *:80>
  ServerName openkm.your-domain.com
  RedirectMatch ^/$ /OpenKM

  ProxyPass /OpenKM ajp://127.0.0.1:8009/OpenKM keepalive=On
  ProxyPassReverse /OpenKM http://openkm.your-domain.com/OpenKM

  ErrorLog /var/log/apache2/openkm-error.log
  CustomLog /var/log/apache2/openkm-access.log combined
</VirtualHost>
```



Change openkm.your-domain.com with your server IP or your domain value.

The VirtualHost ServerName must be other than ServerName in the main Apache configuration. Enable this site configuration:

```
$ cd /etc/apache2/sites-available/
$ sudo a2ensite openkm.conf
```

Check the configuration

Restart Apache:

```
$ sudo /etc/init.d/apache2 restart
```

Check the access your OpenKM installation from <http://openkm.your-domain.com/>.



Another advantage of using Apache is that you can log OpenKM access and generate web statistics.



To check the configuration files can execute the command:

```
/usr/sbin/apache2 -t
```

Red Hat and CentOS

Use the yum application manager to install Apache:

```
$ sudo yum install httpd
```

Enable it at boot:

```
$ sudo chkconfig httpd --level 2345 on
```

Now create the file `/etc/httpd/conf.d/openkm.conf` with this content:

```
$ vim /etc/httpd/conf.d/openkm.conf
```

```
<VirtualHost *:80>
    ServerName openkm.your-domain.com
    RedirectMatch ^/$ /OpenKM

    ProxyPass /OpenKM ajp://127.0.0.1:8009/OpenKM keepalive=On
    ProxyPassReverse /OpenKM http://openkm.your-domain.com/OpenKM

    ErrorLog /var/log/httpd/openkm-error.log
    CustomLog /var/log/httpd/openkm-access.log combined
</VirtualHost>
```



Change `openkm.your-domain.com` with your server IP or your domain value.

Check the configuration

Restart Apache:

```
$ sudo /etc/init.d/httpd restart
```

Check the access your OpenKM installation from <http://openkm.your-domain.com/>.



Another advantage of using Apache is that you can log OpenKM access and generate web statistics.

Windows

Install de apache webserver from <http://httpd.apache.org/download.cgi> (windows binary no mod_ssl)

Edit the `C:\Program files\Apache Software Foundation\Apache2.2\conf\httpd.conf`

```
# Enable proxy modules
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so

#Add in bottom
NameVirtualHost *:80
<VirtualHost *:80>
    ServerName openkm.your-domain.com
    RedirectMatch ^/$ /OpenKM
```

```
ProxyRequests Off
ProxyVia On

<Proxy *>
    AddDefaultCharset off
    Order deny,allow
    Allow from all
    Deny from All
</Proxy>

ProxyPass /OpenKM ajp://127.0.0.1:8009/OpenKM keepalive=On
ProxyPassReverse /OpenKM http://openkm.your-domain.com/OpenKM

ErrorLog logs/openkm-error_log
CustomLog logs/openkm-access_log common
</VirtualHost>
```



Change openkm.your-domain.com with your server ip or your domain value.

Max OS X

Edit the file called `/etc/apache2/apache2.conf` and configure a **ServerName**, enable proxy modules and **mod_proxy**:

```
ServerRoot "/usr"
ServerName "openkm.your-domain.com"

LoadModule proxy_module libexec/apache2/mod_proxy.so
LoadModule proxy_ajp_module libexec/apache2/mod_proxy_ajp.so

<IfModule mod_proxy.c>
    ProxyRequests Off
    <Proxy "*">
        AddDefaultCharset off
        Allow from all
        Deny from all
        Order Deny,Allow
    </Proxy>
    ProxyVia On
</IfModule>
```



Change openkm.your-domain.com with your server IP or your domain value.

Now create the configuration file `/etc/apache2/sites/openkm.conf` with this content:

```
<VirtualHost *:80>
    ServerName openkm.your-domain.com
    RedirectMatch ^/$ /OpenKM

    ProxyPass /OpenKM ajp://127.0.0.1:8009/OpenKM keepalive=On
    ProxyPassReverse /OpenKM http://openkm.your-domain.com/OpenKM

    ErrorLog /var/log/apache2/openkm-error.log
    CustomLog /var/log/apache2/openkm-access.log combined
</VirtualHost>
```

Check the configuration

Restart Apache:

Check the access to your OpenKM installation from <http://openkm.your-domain.com/>.

Another advantage of using Apache is that you can log OpenKM access and generate web statistics.

Troubleshooting

If you see an error message like:



Invalid command 'RewriteEngine', perhaps misspelled or defined by a module not included in the server configuration

You need to enable this Apache module:

```
$ sudo a2enmod rewrite
$ sudo a2enmod proxy_http
$ sudo a2enmod headers
```

Additional information

- [Apache Module mod_proxy](#)
- [Apache Module mod_proxy_ajp](#)
- [Apache Module mod_rewrite](#)
- [mod_rewrite, a beginner's guide](#)
- [Using Apache Virtual Hosts and ProxyPass Together](#)

Configuring Apache HTTP for changing context URL

First of all you must have one of these configurations running on your server:

- [Configuring Apache HTTP Reverse-Proxy](#)

[Configuring Apache HTTPS Reverse-Proxy](#)



On the sample below the <http://openkm.your-domain.com/OpenKM/frontend> has been rewrited to <http://openkm.yourdomain.com/frontend>

Enable the headers module:

```
$ sudo a2enmod headers
```

To change /OpenKM application context try this as your VirtualHost configuration:

```
<VirtualHost *:80>
    ServerName openkm.your-domain.com

    ProxyPass / ajp://localhost:8009/OpenKM/
    ProxyPassReverse / http://openkm.your-domain.com/OpenKM/

    RequestHeader edit X-GWT-Module-Base ^(http://openkm.your-domain.com)/frontend/(
.*)$ $1/OpenKM/frontend/$2
    ProxyPassReverseCookiePath /OpenKM /

    ErrorLog /var/log/apache2/openkm-error.log
    CustomLog /var/log/apache2/openkm-access.log combined
</VirtualHost>
```

You need also set the configuration parameter **system.apache.request.header.fix** to on (true).

Additional information

- [Apache configuration for GWT applications](#)

Configuring Apache HTTPS Reverse-Proxy

Exposing OpenKM directly from Tomcat can be dangerous if you need the application to be accessed from Internet (for example <https://issues.jboss.org/browse/JBAS-3861>). As result you can be infected by [PerlBot](#). Also this 8080 may be blocked by a firewall. For these reasons, it is a good practice to expose your OpenKM installation through the standard web port 80. In the following steps we explain how to configure Apache to handle these request and forward to Tomcat application server using the AJP13 protocol.

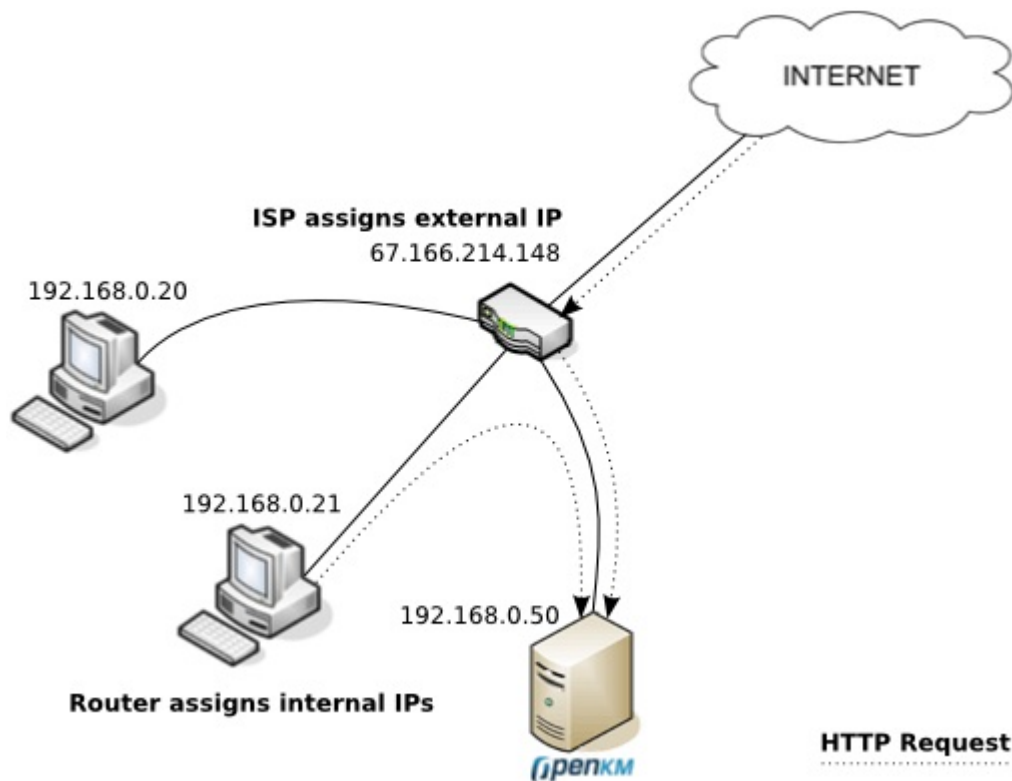
From the Apache documentation: The AJP13 protocol is packet-oriented. A binary format was presumably chosen over the more readable plain text for performance reasons. The web server communicates with the servlet container over TCP connections. To cut down on the expensive process of socket creation, the web server will attempt to maintain persistent TCP connections to the servlet container, and to reuse a connection for multiple request/response cycles.

Internal IP vs external IP

Your OpenKM can be accessed from two different zones: Internet and LAN. This means that to access this server you need to use two IPs: external IP (Internet) and internal IP (LAN).

The internal IP address (also known as "local IP address") is the address that is assigned by your local network router that often begins with 192.168.x.x. These IP addresses can only be seen by other computers in your local network (LAN) and not by any computer connected in an external network such the Internet.

To reach the Internet or a computer in another network your computer is often assigned an external IP address, which can then be used to refer to the computer in your local network.



In the above picture, there are three computers in the local network that have each been assigned their own internal IP address by the router. The ISP is connected to the router and gives the router an external IP address that allows it to communicate with the Internet. On the Internet everyone sees your external IP address, but any information coming from the router is "converted" from the external IP address to the internal IP address.

So if you want your OpenKM installation to be accessible from both LAN and Internet, the trick here is to configure the client computers to resolve your **internal IP** (192.168.0.50) if they are inside the LAN or the **external IP** if they are on Internet. To resolve the Internet IP (67.166.214.148) your computer uses the public DNS. So, you need to configure a sort of DNS server inside the LAN or modify every client [host file](#) to resolve to the internal IP.

Debian and Ubuntu

The first thing in to install the required Apache software. From Debian / Ubuntu you can install Apache with a single command:

```
$ sudo apt-get install apache2
```

Edit the file named `/etc/apache2/apache2.conf` and configure a `ServerName` to prevent warnings in the Apache startup process:

```
$ vim /etc/apache2/apache2.conf
```

```
ServerRoot "/etc/apache2"
ServerName "your-domain.com"
```




Edit apache2.conf is optional.

Enable the proxy module:

```
$ sudo a2enmod proxy_ajp
```

Create the Certificates

```
$ sudo mkdir /etc/apache2/ssl
$ sudo /usr/sbin/make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/apache2/ssl/apache.pem
$ sudo a2enmod ssl
```



If you need to convert .pem file to crt use the command:

```
$ openssl x509 -in cacert.pem -out cacert.crt
```



Also you could be interested on [Configuring OpenSSL](#).

Ensure ports **443** is listen in **/etc/apache2/ports.conf**:

```
$ cat /etc/apache2/ports.conf | grep Listen
```

Create the configuration file **/etc/apache2/sites-available/openkm.conf** with this content:

```
$ vim /etc/apache2/sites-available/openkm.conf
```

```
<VirtualHost *:443>
    ServerName openkm.your-domain.com
    RedirectMatch ^/$ /OpenKM

    ProxyPass /OpenKM ajp://127.0.0.1:8009/OpenKM
    ProxyPassReverse /OpenKM https://openkm.your-domain.com/OpenKM

    ErrorLog /var/log/apache2/openkm.your-domain.com-error.log
    CustomLog /var/log/apache2/openkm.your-domain.com-access.log combined

    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/apache.pem
</VirtualHost>
```



Change openkm.your-domain.com with your server IP or your domain value.

The VirtualHost ServerName must be other than ServerName in the main Apache configuration. Enable this site configuration:

```
$ cd /etc/apache2/sites-available/
$ sudo a2ensite openkm.conf
```

Redirect all HTTP connections to HTTPS

This step is optional.

Edit the configuration file `/etc/apache2/sites-available/openkm.conf` and add these content

```
$ vim /etc/apache2/sites-available/openkm.conf
```

```
<VirtualHost *:80>
  ServerName openkm.your-domain.com
  Redirect permanent / https://openkm.your-domain.com/
</VirtualHost>
```

Check the configuration

Restart Apache:

```
$ sudo /etc/init.d/apache2 restart
```

Check the access your OpenKM installation from <https://openkm.your-domain.com/>.



Another advantage of using Apache is that you can log OpenKM access and generate web statistics.



To check the configuration files can execute the command:

```
$ /usr/sbin/apache2 -t
```

Red Hat and CentOS

Use the yum application manager to install Apache:

```
$ sudo yum install httpd
```

Enable it at boot:

```
$ sudo chkconfig httpd --level 2345 on
```

Create the Certificate

```
$ sudo yum install mod_ssl openssl crypto-utils
```

Generate private keys (for more information visit [Apache HTTP Secure Server Configuration](#))

```
$ genkey your-domain.com
```



Usually you want to generate self-certificate and not sending it to Certify Authority. Is good practice to set a

password on private key, but in this case each time you restart Apache service it'll be demanded.



Also you could be interested on [Configuring OpenSSL](#).

During the process will be generated two files at:

```
ReplaceSSLCertificateFile /etc/pki/tls/certs/your-domain.com.cert
SSLCertificateKeyFile /etc/pki/tls/private/your-domain.com.key
```

Now create the file **/etc/httpd/conf.d/openkm.conf** with this content:

```
$ vim /etc/httpd/conf.d/openkm.conf
```

```
<VirtualHost *:443>
  ServerName openkm.your-domain.com
  ProxyPass / ajp://localhost:8009/OpenKM/
  ProxyPassReverse / http://your-domain.com/OpenKM/
  ErrorLog /var/log/your-domain.com-error.log
  CustomLog /var/log/your-domain.com-access.log combined

  SSLEngine on
  SSLCertificateFile /etc/pki/tls/certs/your-domain.com.cert
  SSLCertificateKeyFile /etc/pki/tls/private/your-domain.com.key
</VirtualHost>
```



Change openkm.your-domain.com with your server ip or your domain value.

Finally must modify **SSLCertificateFile** and **SSLCertificateKeyFile** values in file **/etc/httpd/conf.d/ssl.conf**:

```
$ vim /etc/httpd/conf.d/ssl.conf
```

```
SSLCertificateFile /etc/pki/tls/certs/your-domain.com.cert
SSLCertificateKeyFile /etc/pki/tls/private/your-domain.com.key
```

Check the configuration

Restart Apache:

```
$ sudo /etc/init.d/httpd restart
```

Check the access your OpenKM installation from <https://openkm.your-domain.com/>.



Another advantage of using Apache is that you can log OpenKM access and generate web statistics.

Troubleshooting

If you see an error like:



Invalid command 'RewriteEngine', perhaps misspelled or defined by a module not included in the server configuration

You need to enable this Apache module:

```
$ sudo a2enmod rewrite  
$ sudo a2enmod proxy_http  
$ sudo a2enmod headers
```

Additional information

- [Apache Module mod_proxy](#)
- [Apache Module mod_proxy_ajp](#)
- [Apache Module mod_rewrite](#)
- [mod_rewrite, a beginner's guide](#)
- [Using Apache Virtual Hosts and ProxyPass Together](#)
- [Rewrite HTTP To HTTPS](#)
- [Redirect to SSL](#)
- [Configuración SSL de Tomcat](#)
- [How To Create a SSL Certificate on Apache for CentOS 6](#)

Configuring Nginx HTTP Reverse-Proxy

Exposing OpenKM directly from Tomcat can be dangerous if you need the application to be accessed from the Internet (for example <https://issues.jboss.org/browse/JBAS-3861>). As result it can be infected by [PerlBot](#). Also this 8080 may be blocked by a firewall. For these reasons, is a good idea to expose your OpenKM installation through the standard web port 80. In the following steps we explain how to configure Apache to handle these requests and forward them to Tomcat application server using the AJP13 protocol.

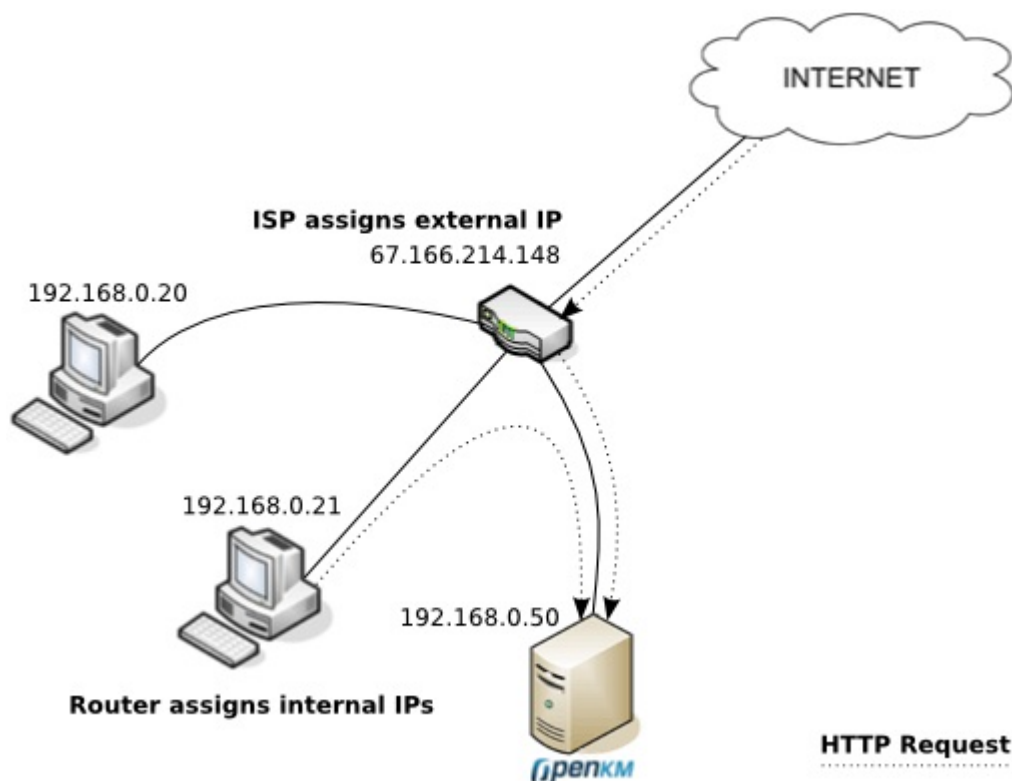
From the Apache documentation: The AJP13 protocol is packet-oriented. A binary format was presumably chosen over the more readable plain text for reasons of performance. The web server communicates with the servlet container over TCP connections. To cut down on the expensive process of socket creation, the web server will attempt to maintain persistent TCP connections to the servlet container, and to reuse a connection for multiple request/response cycles.

Internal IP vs external IP

Your OpenKM can be accessed from two different zones: Internet and LAN. This means that to access this server you need to use two IPs: external IP (Internet) and internal IP (LAN).

The internal IP address (also known as "local IP address") is the address that is assigned by your local network router that often begins with 192.168.x.x. These IP addresses can only be seen by other computers in your local network (LAN) and not by any computers connected in an external network such the Internet.

To reach the Internet or a computer in another network your computer is often assigned an external IP address, which can then be used to refer to the computer in your local network.



In the above picture, there are three computers in the local network that have each been assigned their own internal IP address by the router. The ISP is connected to the router and gives the router an external IP address that allows it to communicate with the Internet. On the Internet everyone sees your external IP address, but any information coming from the router is "converted" from the external IP address to the internal IP address.

So if you want your OpenKM installation accessible from both LAN and Internet, the trick here is configure the client computers to resolve your **internal IP** (192.168.0.50) if they are inside the LAN or the **external IP** if they are on the Internet. To resolve the Internet IP (67.166.214.148) your computer uses the public DNS. So, you need to configure a sort of DNS server inside the LAN or modify every client [host file](#) to resolve to the internal IP.

Debian and Ubuntu

The first thing in to install the required Apache software. From Debian / Ubuntu you can install Apache with a single command:

```
$ sudo apt-get install nginx
```

Now create the configuration file `/etc/nginx/sites-available/openkm` with this content:

```
$ vim /etc/nginx/sites-available/openkm
```

```
server {
    listen 80;
    server_name openkm.your-domain.com;
    # Avoid checking files size #
}
```

```
client_max_body_size 0;

rewrite ^/$ /OpenKM/ permanent;

location /OpenKM/ {
    proxy_set_header X-Forwarded-Host $host;
    proxy_set_header X-Forwarded-Server $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_pass http://localhost:8080/OpenKM/;
}
```



Change `openkm.your-domain.com` with your server ip or your domain value.

Enable this site configuration:

```
$ ln -s /etc/nginx/sites-available/openkm /etc/nginx/sites-enabled/
```

Check the configuration

Restart Nginx:

```
$ sudo /etc/init.d/nginx restart
```

Check the access your OpenKM installation from <http://openkm.your-domain.com/>.

Another advantage of using Nginx is that you can log OpenKM access and generate web statistics.

Red Hat and CentOS

Use the yum application manager to install Nginx:

```
$ sudo yum install nginx
```

Enable it at boot:

```
$ sudo chkconfig nginx --level 2345 on
```

Depending on your CentOS version this Nginx version may be too old so it's better to install it from their repositories. To do this, create a file called `/etc/yum.repos.d/nginx.repo` and include one on these configurations:

CentOS

```
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/centos/$releasever/$basearch/
gpgcheck=0
enabled=1
```

RedHat

```
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/rhel/$releasever/$basearch/
gpgcheck=0
enabled=1
```

Now create the configuration file `/etc/nginx/sites-available/openkm` with this content:

```
$ vim /etc/nginx/conf.d/openkm.conf
```

```
server {
    listen 80;
    server_name openkm.your-domain.com;
    # Avoid checking files size #
    client_max_body_size 0;

    rewrite ^/$ /OpenKM permanent;

    location /OpenKM {
        proxy_set_header X-Forwarded-Host $host;
        proxy_set_header X-Forwarded-Server $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_pass http://localhost:8080/OpenKM;
    }
}
```



Change `openkm.your-domain.com` with your server IP or your domain value.

Check the configuration

Restart Nginx:

```
$ sudo /etc/init.d/nginx restart
```

Check the access your OpenKM installation from <http://openkm.your-domain.com/>.

If you can't access server port from other machines, please take a look at your firewall configuration. For more info read [How To Set Up a Basic Iptables Firewall on Centos 6](#).



Another advantage of using Nginx is that you can log OpenKM access and generate web statistics.



In case you see this kind of errors in the Nginx log:

```
connect() to 127.0.0.1:8080 failed (13: Permission denied)
```

Execute this command and try again:


```
$ setsebool httpd_can_network_connect true
```

Configuring Lucene Analyzer

Depending on the language used in the documents and properties, you have obtain better search results configuring a proper Lucene Analyzer.

By default, OpenKM use the **org.apache.lucene.analysis.standard.StandardAnalyzer** which works fine with English and most languages, but you can get better search results configuring more specific analyzer for you language.



Some analyzers:

- [org.apache.lucene.analysis.en.EnglishAnalyzer](#)
- [org.apache.lucene.analysis.es.SpanishAnalyzer](#)
- [org.apache.lucene.analysis.fr.FrenchAnalyzer](#)
- [org.apache.lucene.analysis.it.ItalianAnalyzer](#)
- [org.apache.lucene.analysis.de.GermanAnalyzer](#)
- [org.apache.lucene.analysis.el.GreekAnalyzer](#)
- [org.apache.lucene.analysis.hi.HindiAnalyzer](#)

More information at:

- [Analyzer](#)
- [ReusableAnalyzerBase](#)
- [StopwordAnalyzerBase](#)



If you are working with oriental languages like Chinese or Japanese you have several analyzers to use. Read [Lucene documentation](#). You can also try [ik-analyzer](#)



If you want only a white space tokenized analyzer can try with this one [WhitespaceAnalyzer](#).

If you have not configured the search analyzer before the first time you start OpenKM, then Lucene indexed will be created using this default analyzer.



If you want to change this configuration property after the OpenKM repository has been created you need to **Rebuild Lucene Indexes**.

Once the operation has been completed, the Lucene indexes will be using the new analyzer.

For more information take a look at [Rebuild indexes](#).

Configure an Analyzer

Edit the `$TOMCAT_HOME/OpenKM.cfg` file and add the line:

```
hibernate.search.analyzer=org.apache.lucene.analysis.es.SpanishAnalyzer
```



The changes will take effect after restarting the application.

Configuring OCR engine

OpenKM can work with several OCR engines, for example Tesseract 2.x, Tesseract 3.x, Cuneiform or Abby among others.



OpenKM can be integrated with any OCR engine that can be executed from command line.


Tesseract

Tesseract is an Open Source OCR engine adopted by Goggle. It works really well. The OCR natively can read TIFF documents and has high ratio of recognition with images 300 dpi of resolution and converted to lineart (1 bit color).

The supported image formats are:

- TIFF
- PNG
- JPG
- GIF

Installation

OS	Description
Ubuntu and Debian	<div>\$ apt-get install tesseract-ocr tesseract-ocr-eng</div> <div>  Depending on your locale you would like to install other language file. </div> <p>More information at:</p> <ul style="list-style-type: none"> • https://github.com/tesseract-ocr • https://code.google.com/p/tesseract-ocr/
Red Hat and CentOS	<p>More information at:</p> <ul style="list-style-type: none"> • https://code.google.com/p/tesseract-ocr/wiki/Compiling • https://code.google.com/p/python-tesseract/wiki/HowToCompileForCentos • http://www.vicchiam.com/blog/?p=168
Windows	<p>Download from https://code.google.com/p/tesseract-ocr/ and follow the installation wizard.</p>

Check installation

From command line execute (where `image.jpg` should be an existing image file):

```
$ tesseract image.jpg text
```



The file **image.jpg** is an image file.

If all goes right a file named **text.txt** what contains the extracted text will be created.

Configuration

Go to **Administration** > [Configuration parameters](#) and edit the parameter named **system.ocr**:

OS	
Linux	<pre>/usr/bin/tesseract \${fileIn} \${fileOut}</pre>
	or
	<pre>/usr/bin/tesseract \${fileIn} \${fileOut} -l esp</pre>
	<div> <p>The parameter -l esp indicate must be used the Spanish language support while executing OCR.</p> <p>To see all tesseract options execute:</p> <pre>\$ tesseract</pre> </div>
Windows	<pre>c:\Tesseract-ocr-3.0.2\tesseract.exe \${fileIn} \${fileOut}</pre>
	or
	<pre>c:\Tesseract-ocr-3.0.2\tesseract.exe \${fileIn} \${fileOut} -l spa</pre>
	<p>The parameter -l esp indicate must be used the Spanish language support while executing OCR.</p> <p>To see all tesseract options execute:</p> <pre>c:\Tesseract-ocr-3.0.2> tesseract.exe</pre>
<p>Parameter \${fileIn} will be replaced internally by image document to be processed.</p> <p>Parameter \${fileOut} will be replaced internally by temporal text file.</p> <p>Can set other additional parameters like -l spa parameter in example.</p> <div> <p>Each parameter is separated by a single "white space".</p> </div>	

Additional information

- <http://code.google.com/p/tesseract-ocr/>
- [Tesseract - Summary & first experiences.](#)
- [Tesseract OCR Google Groups](#)
- [First Interactions with Tesseract OCR on Ubuntu Linux](#)
- <http://code.google.com/p/tesseract-ocr/wiki/ReadMe>
- <http://code.google.com/p/tesseract-ocr/wiki/FAQ>

Cuneiform



We recommend the use of Tesseract rather than cuneiform.

The supported image formats are:

- TIFF
- PNG
- JPG
- GIF

Installation

OS	Description
Ubuntu and Debian	<div>\$ apt-get install cuneiform</div> <p>More information at:</p> <ul style="list-style-type: none"> • http://cognitiveforms.com/products_and_services/cuneiform
Red Hat and Centos	<p>More information at:</p> <ul style="list-style-type: none"> • http://cognitiveforms.com/products_and_services/cuneiform
Windows	<div> <p>Not available to be executed from command line.</p> </div> <p>More information at:</p>

- http://cognitiveforms.com/products_and_services/cuneiform

Check installation

From command line execute (where `image.jpg` should be an existing image file):

```
$ cuneiform image.jpg -o text.txt
```



The file **image.jpg** is an image file.

If all goes right a file named **text.txt** what contains the extracted text will be created.

Configuration

Go to **Administration** > [Configuration parameters](#) and edit the parameter named **system.ocr**:

OS	
Linux	<pre>/usr/bin/cuneiform \${fileIn} -o \${fileOut}</pre> <p>or</p> <pre>/usr/bin/cuneiform \${fileIn} -o \${fileOut} -l esp</pre> <div> The parameter -l esp indicate must be used the Spanish language support while executing OCR. To see all cuneiform options execute: <pre>\$ cuneiform</pre> </div>
Windows	Not available.

Parameter **\${fileIn}** will be replaced internally by image document to be processed.

Parameter **\${fileOut}** will be replaced internally by temporal text file.

Can set other additional parameters like **-l spa** parameter in example.

Before the **\${fileOut}** must be set **-o** parameter.
 Each parameter is separated by a single "white space".

Abby OCR for linux

OCR4 Linux is a commercial OCR engine. It is the slowest of all tested tools, but keep in mind that it also reads nearly any image format, while you probably need to convert your images for the other tools first.

Installation

OS	Description
Ubuntu, Debian, Red Hat and Centos	More information at: <ul style="list-style-type: none"> http://www.ocr4linux.com/
Windows	Not available.

Check installation

From command line execute (where *image.jpg* should be an existing image file):

```
$ /usr/local/bin/abbyocr9 abbyocr11 -ii -fm -if image.jpg -tet UTF8 -of text.txt
```



The file **image.jpg** is an image file.

If all goes right will be created a file named **text.txt** what contains the extracted text.

Configuration

Go to **Administration** > [Configuration parameters](#) and edit the parameter named **system.ocr**:

OS	
Linux	<pre>/usr/local/bin/abbyocr11 -ii -fm -if \${fileIn} -tet UTF8 -of \${fileOut}</pre> <p>There're a lot of parameters, to see all ocr4linux options execute:</p> <pre>\$ abbyocr11 -?</pre>
Windows	Not available.

Parameter **\${fileIn}** will be replaced internally by image document to be processed.

Parameter **\${fileOut}** will be replaced internally by temporal text file.

Can set other additional parameters like **-l spa** parameter in example.



Each parameter is separated by a single "white space".

Dictionary

You can also use an OpenOffice.org dictionary to enhance the OCR process. Can find these language specific dictionaries at [OpenOffice.org Dictionary Repository](#).

Configuration

Go to **Administration** > [Configuration parameters](#) and edit the parameter named **system.openoffice.dictionary**:

OS	
Linux	<code>/home/openkm/dictionary/en-GB.zip</code>
Windows	<code>c:\tomcat-7.0.61\dictionary\en-GB.zip</code>



Dictionaries comes in .zip or oxt format. Can use both.

OCR rotate configuration

There's a configuration parameter that force document ocr on rotated document.



The feature is useful for example in upside down scanned pages, where standard OCR direction will fails.

Configuration

Go to **Administration** > [Configuration parameters](#) and edit the parameter named **system.ocr.rotate**:

Property	Description
system.ocr.rotate	<p>The parameter is a collection of degrees separated by character ";"</p> <pre>0;90;180;270;</pre>

Additional information:

- [Linux OCR Software Comparison](#)
- [Tesseract OCR](#)
- [Cuneiform](#)

- [*Ocrop*](#)
- [*Gocr*](#)
- [*Ocrad*](#)

Configuring OpenSSL

Create OpenSSL configuration file



This sample has been done with Ubuntu and Debian server, for Red Hat and CentOS the path of Apache files is not the same.

On Ubuntu and Debian the Apache path `/etc/apache2` is `/etc/httpd` on Red Hat and CentOS.

Edit file `/etc/ssl/openssl.cnf`

```
$ vim /etc/ssl/openssl.cnf
```

```

RANDFILE          = $ENV::SSLDIR/.rnd

[ ca ]
default_ca        = CA_default

[ CA_default ]
dir               = /opt/ca
certs             = $dir/certs
new_certs_dir    = $dir/newcerts
crl_dir           = $dir/crl
database         = $dir/index.txt
private_key       = $dir/private/ca.key
certificate       = $dir/ca.crt
serial           = $dir/serial
crl               = $dir/crl.pem
RANDFILE         = $dir/private/.rand
default_days      = 365
default_crl_days  = 30
default_md        = sha1
preserve         = no
policy            = policy_anything
name_opt          = ca_default
cert_opt          = ca_default

[ policy_anything ]
countryName       = optional
stateOrProvinceName = optional
localityName      = optional
organizationName  = optional
organizationalUnitName = optional
commonName        = supplied
emailAddress       = optional

[ req ]
default_bits      = 2048
default_md        = sha1
default_keyfile    = privkey.pem
distinguished_name = req_distinguished_name
x509_extensions    = v3_ca
string_mask        = nombstr

[ req_distinguished_name ]
countryName          = Country Name (2 letter code)
countryName_min      = 2
countryName_max       = 2
stateOrProvinceName  = State or Province Name (full name)

```

```

localityName           = Locality Name (eg, city)
0.organizationName     = Organization Name (eg, company)
organizationalUnitName = Organizational Unit Name (eg, section)
commonName             = Common Name (eg, YOUR name)
commonName_max         = 64
emailAddress           = Email Address
emailAddress_max       = 64

[ usr_cert ]
basicConstraints       = CA:FALSE
# nsCaRevocationUrl    = https://url-to-exposed-clr-list/crl.pem

[ ssl_server ]
basicConstraints       = CA:FALSE
nsCertType            = server
keyUsage              = digitalSignature, keyEncipherment
extendedKeyUsage       = serverAuth, nsSGC, msSGC
nsComment              = "OpenSSL Certificate for SSL Web Server"

[ ssl_client ]
basicConstraints       = CA:FALSE
nsCertType            = client
keyUsage              = digitalSignature, keyEncipherment
extendedKeyUsage       = clientAuth
nsComment              = "OpenSSL Certificate for SSL Client"

[ v3_req ]
basicConstraints       = CA:FALSE
keyUsage              = nonRepudiation, digitalSignature, keyEncipherment

[ v3_ca ]
basicConstraints       = critical, CA:true, pathlen:0
nsCertType            = sslCA
keyUsage              = cRLSign, keyCertSign
extendedKeyUsage       = serverAuth, clientAuth
nsComment              = "OpenSSL CA Certificate"

[ crl_ext ]
basicConstraints       = CA:FALSE
keyUsage              = digitalSignature, keyEncipherment
nsComment              = "OpenSSL generated CRL"

```

Create Certification Authority (CA)

Create folder structure to hold certificates:

```

$ mkdir /opt/met-ca
$ mkdir /opt/met-ca/certs
$ mkdir /opt/met-ca/crl
$ mkdir /opt/met-ca/newcerts
$ mkdir /opt/met-ca/private
$ mkdir /opt/met-ca/requests
$ touch /opt/met-ca/index.txt
$ echo "01" > /opt/ca/serial
$ echo "01" > /opt/ca/crlnumber
$ chmod 700 /opt/ca

```

Create public and private auto signed keys of our own Certification Authority (in the example is valid for 10 years). We will need to specify certificate data, password to crypt and the process will generate two files:

```
$ openssl req -config /etc/ssl/openssl.cnf -new -x509 -days 3650 -sha1 -newkey rsa:2048 -keyout /opt/ca/private/ca.key -out /opt/ca/ca.crt
```

Change key grants:

```
$ chmod 600 /opt/met-ca/private/ca.key
```

Create Apache Certification

Create a pair of keys. We will need to specify password and certification data. Will be created the file key (server.key) and certification request (server.pem).

```
$ openssl req -new -sha1 -newkey rsa:2048 -nodes -keyout server.key -out server.pem
```

CA should sign the certification request. Copy server.pem to certification request and sign.

```
$ openssl ca -config /etc/ssl/openssl.cnf -policy policy_anything -extensions ssl_server -out requests/server-signed.pem -infiles requests/server.pem
```

Apache will be waiting a PEM format, is mandatory convert to this format:

```
$ openssl x509 -in requests/server-signed.pem -out requests/server.crt
```

Apache configuration

Copy server.crt and server.key into /etc/apache2/ssl (if SSL folder not exists, create it). Change files grants (server.key should only be read by root user, server.crt should have read grant to everybody):

```
$ chmod 400 server.key  
$ chmod 444 server.crt
```

If Apache is not listening SSL port, enable it configuring /etc/apache2/ports.conf:

```
Listen 80  
Listen 443
```

Enable Apache SSL support:

```
$ apt-get install libapache2-mod-ssl
```

Indicate Apache to load SSL modules:

```
$ a2enmod ssl
```

Configure website:

Link default-ssl file to enable SSL:

```
$ ln -s /etc/apache2/sites-available/default-ssl /etc/apache2/sites-enabled/default-ssl
```

Edit /etc/apache2/sites-availables/default-ssl (enable SSL engine and set SSLCertificateFile and SSLCertificateKeyFile)

```
SSLEngine on
SSLCertificateFile /etc/apache2/ssl/server.crt
SSLCertificateKeyFile /etc/apache2/ssl/server.key
```

Create user certificate:

To create certificates and sign with CA, first create the pair of keys. We will need to specify password and certification data.

Will be created a eky (user.key) and certification request (user.pem):

```
$ openssl req -new -sha1 -newkey rsa:2048 -nodes -keyout user.key -out user.pem
```

With CA is needed sign certificates:

```
$ cp /tmp/usuario.pem /opt/ca/requests
$ openssl ca -config /etc/ssl/openssl.cnf -policy policy_anything -extensions [b]ssl_client[/b] -out requests/user-signed.pem -
infiles requests/user.pem
```

Convert certificate to PKCS12 to be recognized by browser

```
$ openssl pkcs12 -export -clcerts -in user-signed.pem -inkey usuario.key -out user.p12
```

The files user-signed.pem, user.pem and user.key can be deleted

```
$ wipe user-signed.pem user.key user.pem
```



The **user.p12** should be installed to allow user browser.

Certification revocation

If the server has been compromised the CA should be revoked. To revoke a certificate should see serial number at /opt/ca/index.txt.

```
$ openssl ca -config /etc/ssl/openssl.cnf -revoke newcerts/<serial-number>.pem
```

Creat a list of revoked certificates (CRL)

```
$ openssl ca -config /etc/ssl/openssl.cnf -gencrl -crlexts crt_ext -md sha1 -out crt.pem
```

To send to the browser list should be converted to DER format:

```
$ openssl crl -in crt.pem -out ca.crl -outform DER
```

Now can be sent to browser.

Add CRL to Apache

Copy CRL file to Apache SSL folder:

```
$ cp /tmp/crt.pem /etc/apache2/ssl/ca.crl
```

Modify /etc/apache2/sites-available/default-ssl file, add SSLCARevocationFile:

```
SSLCARevocationFile /etc/apache2/ssl/ca.crl
```

Restart Apache service.

Configuring SWFTools

By default the application comes with tomcat bundle with pre-configured SWFTools into.



In most cases it is not necessary install SWFTools in your server. Only in case it presents some problem with pre-configured SWFTools into tomcat bundle.



Application take advantage of SWFTools to convert PDF document to SWF to be shown into UI preview tab.
See [FlexPaper: Converting Documents](#) for more info.

Ubuntu

```
sudo add-apt-repository ppa:guilhem-fr/swftools
sudo apt-get update
sudo apt-get install swftools
```

Improving performance

For better performance, use this configuration.

Property	Description
system.swftools.pdf2swf	<p>Pdf2swf tool used to convert PDF documents to SWF.</p> <pre>/usr/bin/pdf2swf -T 9 -f -t -G -s storeallcharacters \${fileIn} -o \${fileOut}</pre>

The parameters specifies that:

- **-f** : Fonts should be embedded, improves searchability in the document
- **-t** : Inserts a stop between each frame, improves stability
- **-G** : Makes the document smaller and faster to render
- **-s storeallcharacters** : Stores all character information about the texts in the document, improves searchability

Red Hat and Centos

Better if you compile version 0.9.x

```
yum install zlib-devel libjpeg-devel giflib-devel freetype-devel gcc gcc-c++
wget http://www.swftools.org/swftools-0.9.1.tar.gz
tar xzf swftools-0.9.1.tar.gz
```



```
cd swftools-0.9.1
./configure --prefix=/usr/local
make
make install
```

Improving performance

For better performance, use this configuration.

Property	Description
system.swftools.pdf2swf	<p>Pdf2swf tool used to convert PDF documents to SWF.</p> <pre>/usr/bin/pdf2swf -T 9 -f -t -G -s storeallcharacters \${fileIn} -o \${fileOut}</pre>

The parameters specifies that:

- **-f** : Fonts should be embedded, improves searchability in the document
- **-t** : Inserts a stop between each frame, improves stability
- **-G** : Makes the document smaller and faster to render
- **-s storeallcharacters** : Stores all character information about the texts in the document, improves searchability



Take a look at [pdf2swf wiki](#) for complete parameters reference.

Debian

This package is not include in the Debian repositories, so you need to compile from source:

```
aptitude install build-essential libgif-dev xpdf libfreetype6 libfreetype6-dev libjpeg62 libjpeg8 libjpeg8-dev
wget http://www.swftools.org/swftools-0.9.1.tar.gz
tar xzf swftools-0.9.1.tar.gz
cd swftools-0.9.1
./configure --prefix=/usr/local
make
make install
```



Or you can try to install the Ubuntu package.



Take a look at [pdf2swf wiki](#) for complete parameters reference.

Improving performance

For better performance, use this configuration.

Property	Description
system.swftools.pdf2swf	<p>Pdf2swf tool used to convert PDF documents to SWF.</p> <pre>/usr/bin/pdf2swf -T 9 -f -t -G -s storeallcharacters \${fileIn} -o \${fileOut}</pre>

The parameters specifies that:

- *-f* : Fonts should be embedded, improves searchability in the document
- *-t* : Inserts a stop between each frame, improves stability
- *-G* : Makes the document smaller and faster to render
- *-s storeallcharacters* : Stores all character information about the texts in the document, improves searchability

Windows

Download last Swftools <http://www.swftools.org/download.html> and install it (for example at c:\tomcat-7.0.61\tools\swftools).

Improving performance

For better performance, use this configuration.

Property	Description
system.swftools.pdf2swf	<p>Pdf2swf tool used to convert PDF documents to swf.</p> <pre>c:\tomcat-7.0.61\tools\swftools\pdf2swf.exe -T 9 -f -t -G -s storeallcharacters \${fileIn} -o \${fileOut}</pre>

The parameters specifies that:

- *-f* : Fonts should be embedded, improves searchability in the document
- *-t* : Inserts a stop between each frame, improves stability
- *-G* : Makes the document smaller and faster to render
- *-s storeallcharacters* : Stores all character information about the texts in the document, improves searchability



Take a look at [pdf2swf wiki](#) for complete parameters reference.

Troubleshooting

Conversion problems with big files

If the document is large (has many pages) or is complex, the conversion may fail with this error:



ERROR ID Table overflow

ERROR This file is too complex to render- SWF only supports 65536 shapes at once

You can workaround this error adding the **-s poly2bitmap** parameter which convert graphics to bitmaps:

Property	Description
system.swftools.pdf2swf	<pre>/usr/bin/pdf2swf -T 9 -f -t -G -s poly2bitmap -s storeallcharacters \${fileIn} -o \${fileOut}</pre>

Segmentation fault (core dumped)

This is a log of segmentation fault error:

```
/home/openkm/tomcat-7.0.61/bin/pdf2swf -f -T 9 -t -s storeallcharacters input.pdf -o output.swf
NOTICE processing PDF page 1 (878x1105:0:0) (move:0:0)
NOTICE File contains forms
NOTICE File contains pbm pictures
NOTICE File contains soft masks
WARNING multiply blended transparency groups not yet supported!
NOTICE File contains transparency groups
Segmentation fault (core dumped)
```

You can try the **-s poly2bitmap** or **-s bitmap** arguments. In our experience it seems the bitmap option is more reliable for fixing display issues, but of course you decimate any text, and in some cases increase file size and processing time. You can also add the **-s multiply=4** the bitmap argument to get a really crisp bitmap output further increasing size and processing time, but gets you pretty close to the original PDF.

```
/home/openkm/tomcat-7.0.61/bin/pdf2swf -f -T 9 -t -s bitmap -s storeallcharacters input.pdf -o output.swf
NOTICE processing PDF page 1 (878x1105:0:0) (move:0:0)
NOTICE Writing SWF file output.swf
```

Error with PDF with copy protected

Perhaps the PDF is copy protected. Hence, you're not allowed to convert it.



FATAL PDF disallows copying

If you created this PDF using Acrobat Distiller, try switching the copy protection checkbox off. If you can't disable the protection you can modify the 'pdf2swf' source code to disable this check (see <http://www.foolabs.com/xpdf/cracking.html>).

The pre-configured pdf2swf tool into tomcat bundle comes with this hacking.

Configuring mail

The application sometimes sends emails to users, for example when the users have been subscribed to a document or folder.



By default tomcat mail is configured assuming your server has a mail service binding in your localhost.

This is the default value in the `$TOMCAT_HOME/conf/server.xml` file.

```
<Resource auth="Container" name="mail/OpenKM" type="javax.mail.Session"
    mail.from="testing@openkm.com" mail.smtp.host="localhost" />
```

Change mail configuration

Get your mail account data (same than you will use to configure standard mail client, like Outlook or Thunderbird).

Edit the file `$TOMCAT_HOME/conf/server.xml`:

```
vim /home/openkm/tomcat/conf/server.xml
```

After editing the `server.xml` should looks something like this:

```
<Resource name="mail/OpenKM" type="javax.mail.Session"
    mail.transport.protocol="smtp"
    mail.smtp.auth="true"
    mail.smtp.host="your smtp server"
    mail.smtp.user="<username>"
    password="<password>"
    mail.port="25"
    mail.debug="true"/>
```

Gmail example

```
<Resource name="mail/OpenKM" auth="Container" type="javax.mail.Session"
    mail.transport.protocol="smtp"
    mail.smtp.auth="true"
    mail.smtp.host="smtp.gmail.com"
    mail.smtp.port="465"
    mail.smtp.user="<username>"
    password="<password>"
    mail.from="<sender email>"
    mail.smtp.quitwait="false"
    mail.smtp.starttls.enable="true"
    mail.smtp.socketFactory.class="javax.net.ssl.SSLSocketFactory"
    mail.debug="true"/>
```

Office 365

```
<Resource name="mail/OpenKM" auth="Container" type="javax.mail.Session"
    mail.transport.protocol="smtp"
    mail.smtp.starttls.enable="true"
```

```
mail.smtp.auth="true"  
mail.smtp.host="smtp.office365.com"  
mail.smtp.port="587"  
mail.smtp.user="<username>"  
mail.from="<username>"  
password="<password>"  
mail.debug="true" />
```

Additional information



Optionally, you can use these attributes :

- `mail.user="<username>"`

If you get problems during tomcat startup, can try using the attribute `mail.smtp.quitwait="false"`.



A complete list of all available parameters can be found at [sun.mail.smtp package Javadoc](#) and [javax.mail package Javadoc](#).

Configuring Tomcat memory utilization

It's also a good idea to configure the Tomcat memory utilization.

*Edit the file `$TOMCAT_HOME/bin/setenv.sh` and edit the parameter **JAVA_OPTS** where you can increase the system memory managed by the JVM (Java Virtual Machine).*

OpenJDK 1.8

This is the default configuration file values. Tomcat will you 2048MB (`Xmx` parameter) and a heap is dinamically reserved by the JVM. For this configuration your server needs at least ≥ 3.5 GB of RAM (Is a good practice always getting 1GB of free RAM for OS).

```
JAVA_OPTS="$JAVA_OPTS -Xms256m -Xmx2048m -XX:+UseConcMarkSweepGC"
JAVA_OPTS="$JAVA_OPTS -Djava.awt.headless=true -Dlog4j.configuration=file://$CATALINA_
JAVA_OPTS="$JAVA_OPTS -Dfile.encoding=UTF-8 -Dsun.jnu.encoding=UTF-8 -Dmail.mime.igno
CATALINA_PID=$CATALINA_HOME/catalina.pid
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$CATALINA_HOME/lib/sigar
```

OpenJDK 1.7

This is the default configuration file values. Tomcat will you 2048MB (`Xmx` parameter) and a maximum heap of 512MB (- `XX:MaxPermSize` parameter). For this configuration your server needs at least ≥ 3.5 GB of RAM (Is a good practice always getting 1GB of free RAM for OS).

```
JAVA_OPTS="$JAVA_OPTS -Xms256m -Xmx2048m -XX:PermSize=256m -XX:MaxPermSize=512m -XX:+
JAVA_OPTS="$JAVA_OPTS -Djava.awt.headless=true -Dlog4j.configuration=file://$CATALINA_
JAVA_OPTS="$JAVA_OPTS -Dfile.encoding=UTF-8 -Dsun.jnu.encoding=UTF-8 -Dmail.mime.igno
CATALINA_PID=$CATALINA_HOME/catalina.pid
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$CATALINA_HOME/lib/sigar
```

Configuring Tomcat port

By default the tomcat binding ports are 8005, 8080 and 8009.



If you have other tomcat running on same server or other application like JBoss Application Server, probably these ports are already used. In this case you should change default ports.

Change tomcat ports



The port change must be done with the tomcat application stopped.

Edit \$TOMCAT_HOME/conf/server.xml and look for all Tomcat port properties.

Service	Default port	Example of new port
Shut-down Port	8005	8105
	Change <pre><Server port="8005" shutdown="SHUTDOWN"></pre> to <pre><Server port="8105" shutdown="SHUTDOWN"></pre>	
Tomcat Connector Port	8080	8180
	Change <pre><Connector address="0.0.0.0" connectionTimeout="20000" port="8080" protocol="HTTP/1.1" redirectPort="8443"/></pre> to <pre><Connector address="0.0.0.0" connectionTimeout="20000" port="8180" protocol="HTTP/1.1" redirectPort="8443"/></pre>	
AJP Connector Port	8009	8109
	Change	


```
<Connector address="127.0.0.1" port="8009" protocol="AJP/1.3"
redirectPort="8443"/>
```

to

```
<Connector address="127.0.0.1" port="8109" protocol="AJP/1.3"
redirectPort="8443"/>
```



To shut-down the different instances execute the **shutdown.sh** script from each `$TOMCAT_HOME/bin`.

Configuring Tomcat session time-out

By default the session time-out is set to 30 minutes.

To change the Tomcat session time-out, edit the `$TOMCAT_HOME/conf/web.xml` and change the values into the session section:

```
<!-- You can set the default session timeout (in minutes) for all newly -->
<!-- created sessions by modifying the value below. -->
<session-config>
    <session-timeout>30</session-timeout>
</session-config>
```

Configuring Tomcat SSL/TLS

Secure Sockets Layer (SSL) and the new Transport Layer Security (TLS), are cryptographic protocols designed to provide a secure communication between two computers. They use certificates and cryptography to authenticate the counterpart with whom they are communicating and to negotiate a session key. This session key is used to encrypt data between the endpoints. A consequence of using certificates, is that Certificate Authorities (CA) and a public key infrastructure are necessary to verify the relation between a certificate and its owner, as well as to generate, sign, and administer the validity of certificates. Due to this reason you have two options to configure SSL:

- **Self-signed Certificate:** it's free because you create your own certificate.
- **CA Certificate:** It cost money because you need to pay for it to a certification authority.

Tomcat currently operates only on JKS, PKCS11 or PKCS12 format keystores. The JKS format is Java's standard "Java KeyStore" format, and is the format created by the key tool command-line utility. This tool is included in the OpenJDK. The PKCS12 format is an Internet standard, and can be manipulated via (among other things) OpenSSL and Microsoft's Key-Manager.

Each entry in a key store is identified by an alias string. Whilst many key store implementations treat aliases in a case non sensitive manner, case sensitive implementations are available. The PKCS11 specification, for example, requires that aliases are case sensitive. To avoid issues related to the case sensitivity of aliases, it is not recommended to use aliases that differ only in case.

Self-signed Certificate

Create the Certificate Key store

```
$ keytool -genkey -alias tomcat -keyalg RSA -keystore self-signed.jks
Enter keystore password: *****
Re-enter new password: *****
What is your first and last name? [Unknown]: Self-Signed Certificate
What is the name of your organizational unit? [Unknown]: System Management
What is the name of your organization? [Unknown]: OpenKM
What is the name of your City or Locality? [Unknown]: Palma de Mallorca
What is the name of your State or Province? [Unknown]: Islas Baleares
What is the two-letter country code for this unit? [Unknown]: ES
Is CN=Self-Signed Certificate, OU=System Management, O=OpenKM, L=Palma de Mallorca, ST=Islas Baleares, C=ES
correct?
[no]: yes
```

Enter key password for <tomcat>

(RETURN if same as keystore password):



The password **changeit** is the default password set into \$TOMCAT_HOME/conf/server.xml file

Edit the Tomcat Configuration file

Enable the SSL connector:

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
maxThreads="150" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="${catalina.home}/sef-signed.jks" keystorePass="changeit" key.
```

Check configuration

Restart Tomcat service and check your OpenKM URL with **https://YOUR_SERVER_IP:8443/OpenKM**.

CA Certificate

Prepare the Certificate Key store

```
$ keytool -genkey -alias tomcat -keyalg RSA -keystore ca-signed.jks
```

Enter keystore password: *****

Re-enter new password: *****

What is your first and last name? [Unknown]: CA-Signed Certificate

What is the name of your organizational unit? [Unknown]: System Management

What is the name of your organization? [Unknown]: OpenKM

What is the name of your City or Locality? [Unknown]: Palma de Mallorca

What is the name of your State or Province? [Unknown]: Islas Baleares

What is the two-letter country code for this unit? [Unknown]: ES

Is CN=Self-Signed Certificate, OU=System Management, O=OpenKM, L=Palma de Mallorca, ST=Islas Baleares, C=ES correct?

[no]: yes

Enter key password for <tomcat>

(RETURN if same as keystore password):



The password **changeit** is the default password set into \$TOMCAT_HOME/conf/server.xml file

Create the Certificate Signing Request

This step is different from the self-signed certificate because you need to send this information to the Certification Authority (CA) which need to validate and sign. Your chosen CA will send you back some files with the Root Certificate the Intermediate Certificates and the Primary Certificate (which is the certificate created for you).

```
$ keytool -certreq -keyalg RSA -alias tomcat -file ca-signed.csr -keystore ca-signed.jks
```

Install the Root Certificate

Every time you install a certificate to the key store you must enter the key store password that you chose when you generated it. Enter the following command to install the Root Certificate file:

```
$ keytool -import -trustcacerts -alias root -file RootCertFileName.crt -keystore ca-signed.jks
```

If you receive a message that says "Certificate already exists in system-wide CA keystore under alias <...> Do you still want to add it to your own keystore? [no]:", select Yes. If successful, you will see "Certificate was added to keystore".

Install the Intermediate Certificate

If your certificate authority provided an intermediate certificate file, you will need to install it here by typing the following command:

```
$ keytool -import -trustcacerts -alias intermediate -file IntermediateCertFileName.crt -keystore ca-signed.jks
```

If successful, you will see "Certificate was added to keystore".

Install the Primary Certificate

Type the following command to install the Primary certificate file (for your domain name):

```
$ keytool -import -trustcacerts -alias tomcat -file PrimaryCertFileName.crt -keystore ca-signed.jks
```



The command `$ keytool -list -keystore ca-signed.jks -v` shows all certificates imported.

Edit the Tomcat Configuration file

Enable the SSL connector:

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
    maxThreads="150" scheme="https" secure="true"
    clientAuth="false" sslProtocol="TLS"
    keystoreFile="${catalina.home}/ca-signed.jks" keystorePass="changeit" keyA
```

Check configuration

Restart Tomcat service and check your OpenKM URL with **`https://YOUR_SERVER_IP:8443/OpenKM`**.

CA Certificate without preparing keystore

Sometimes the person who installs the certificate is not the same that made the request to obtain the certificate. In this case every step is similar to the previous one with the exception of the installation of the Primary Certificate.

In this case, when the primary certificate is installed, it is necessary to install the private RSA key too. To do it, apart from the primary certificate, the private RSA key must be given.

Obtain p12 certificate

Obtain p12 certificate from private key and primary certificate. This is the way that both keys are installed.

```
$ openssl pkcs12 -export -in PrimaryCertFileName.crt -inkey privateRSA.key > PrimaryCertFileName.p12
```

If successful, PrimaryCertFileName.p12 certificate will be created.

Install p12 Certificate

Type the following command to install the .p12 certificate file (for your domain name):

```
$ keytool -v -importkeystore -srckeystore PrimaryCertFileName.p12 -srcstoretype PKCS12 -destkeystore ca-signed.jks -deststoretype JKS
```

If successful, you will see "Certificate was added to keystore".

Change alias to Primary Certificate

Type the following command to install the .p12 certificate file (for your domain name):

```
$ keytool -changealias -keystore ca-signed.jks -alias 1 -destalias tomcat
```

Let's Encrypt

Let's Encrypt uses a client application which automates the certificate generation and renovation.

Client installation

In Linux you can install the tool with these commands:

```
$ wget https://dl.eff.org/certbot-auto  
$ chmod a+x certbot-auto
```

The first will install the required dependencies:

```
$ sudo ./certbot-auto
```

Certificate generation

Now it's time to generate a certificate. This is a sample command line used to generate only the certificate. For more info about usage and command line options, please visit <https://certbot.eff.org/>:

```
$ sudo ./certbot-auto certonly --standalone -m contact@mail-server.com -d sample-domain.com
```

This will create a directory structure where you can locate the certificate. This directory is protected so you need to switch to root to go into:

```
$ sudo su -  
$ cd /etc/letsencrypt/live/sample-domain.com
```

As you can see there are several PEM files. Let's create a PKCS12 certificate from these files which will be used in Tomcat:

```
$ openssl pkcs12 -export -in cert.pem -inkey privkey.pem -CAfile chain.pem -caname root -name tomcat -out mycert.p12
```

Keep the password safe because it will be used later.

Tomcat configuration

Copy the generated file to TOMCAT_HOME and edit the \$TOMCAT_HOME/conf/server.xml file to configure the new certificate:

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"  
    maxThreads="150" scheme="https" secure="true"  
    clientAuth="false" sslProtocol="TLS"  
    keystoreFile="`${catalina.home}/mycert.p12" keystorePass="changeit" keyAlias="tomcat"
```

Once Tomcat is started, the 8443 port will accept SSL connections. But remember these connections should only be done using the domain you used to generate the certificate, otherwise the browser will alert you about the site.

Other configuration

To redirect from non SSL port 8080 to SSL port 8443 edit the \$TOMCAT_HOME/web.xml and append at the end:

```
<security-constraint>  
  <web-resource-collection>  
    <web-resource-name>App_nmae</web-resource-name>  
    <url-pattern>/*</url-pattern>  
    <http-method>GET</http-method>  
    <http-method>POST</http-method>  
  </web-resource-collection>  
  
  <user-data-constraint>  
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>  
  </user-data-constraint>  
</security-constraint>
```

And restart Tomcat.

Known issues

Windows uses a different protocol

In Window systems the protocol used in the Connector must be changed so the final configuration is:

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol" SSLEnable
maxThreads="150" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="${catalina.home}\ca-signed.jks" keystorePass="changeit" keyA
```

ca-bundles certificates

Sometimes the certificates provider gives serveral certificates in one file with .ca-bundle extension. Sometimes, when this bundle is imported as one file, only one of the certicates is imported. In is this case, the solution is to split the bundle in several files (once for each certificate).

To do it, it is possible to edit the ca-bundle file, and each certificate is separated by:

```
-----BEGIN CERTIFICATE-----
.
.
.
.
-----END CERTIFICATE-----
```

Additional information

Are several ways to configure SSL in Tomcat for more information:

- [Apache Tomcat - SSL / TLS Configuration HOW-TO](#)
- [A Simple, Step-By-Step Guide To Apache Tomcat SSL Configuration | MuleSoft](#)

Configuring Tomcat timezone

Tomcat works in GMT timezone if you want to change it you should add something like at `$TOMCAT_HOME/bin/setenv.sh` (or `$TOMCAT_HOME/bin/setenv.bat` if you're on Windows):

```
CATALINA_OPTS="-Duser.timezone=Asia/Hong_Kong"
```

More information about [timezone list](#).

Configuring Tomcat to work behing an Internet proxy



We encourage to consider do not configure Internet connection in Servers behing an Internet proxy.

It have some disadvantages like:

- You will need to use your proxy data each time you will upgrade the application with OpenKM automatic updaters tools.
- It's need to configure JVM parameters with your Internet proxy data for the application be able to connect to Internet.

Is not much clear what are the advantatges for this kind of configuration on Production Servers, what are not the same scenario than end users. **Take it as is, only a suggestion.**

Linux



Configuration below is based on setting proxy configuration parameters in the JVM environment. There's another option what is configuring proxy at OS level, here are several option one of them is:

Edit your `"/etc/bash.bashrc"` file as root and put these lines at the end of your `"/etc/bash.bashrc"` file:

```
export http_proxy=http://username:password@proxyserver.net:port/
export ftp_proxy=http://username:password@proxyserver.net:port/
```

- Stop OpenKM.
- Edit the `$TOMCAT_HOME/bin/setenv.sh` file and add the lines:



The <http://proxy.url.com> value must be your own url host of the proxy.

The `"user"` value must be the user for getting connection to the proxy. If your **proxy does not require authentication** you can remove the entire line where are set these parameters.

The `"password"` values must be the password of the user for getting connection to the proxy. If your **proxy does not require authentication** you can remove the entire line where are set these parameters.

```
set JAVA_OPTS=%JAVA_OPTS% -Dhttp.proxyHost=http://proxy.url.com -Dhttps.proxyHost=htt
set JAVA_OPTS=%JAVA_OPTS% -Dhttp.proxyUser=user -Dhttps.proxyUser=user
set JAVA_OPTS=%JAVA_OPTS% -Dhttp.proxyPassword=password -Dhttps.proxyPassword=password
```

At the end your `setenv.sh` should looking something like that :

```
JAVA_OPTS="$JAVA_OPTS -Xms256m -Xmx2048m -XX:+UseConcMarkSweepGC -Djava.awt.headless=
JAVA_OPTS="$JAVA_OPTS -Dfile.encoding=UTF-8 -Dsun.jnu.encoding=UTF-8 -Dmail.mime.igno
set JAVA_OPTS=%JAVA_OPTS% -Dhttp.proxyHost=http://proxy.url.com -Dhttps.proxyHost=htt
set JAVA_OPTS=%JAVA_OPTS% -Dhttp.proxyUser=user -Dhttps.proxyUser=user
```

```
set JAVA_OPTS=%JAVA_OPTS% -Dhttp.proxyPassword=password -Dhttps.proxyPassword=password
CATALINA_PID=$CATALINA_HOME/catalina.pid
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$CATALINA_HOME/lib/sigar
```



If you have issues resolving DNS you can also try with this configuration what is trying to use Google DNS:

```
set JAVA_OPTS=%JAVA_OPTS% -Dsun.net.spi.nameservice.nameservers=8.8.8.8
set JAVA_OPTS=%JAVA_OPTS% -Dsun.net.spi.nameservice.provider.1=dns,sun
```

- **Start OpenKM.**
- Go to **Administration > Tools > Scripting** and execute the script below (should be shown google html code).

```
import java.net.URL;
import java.net.HttpURLConnection;
import java.io.BufferedReader;

URL url = new URL("http://www.google.com");
HttpURLConnection urlConn = (HttpURLConnection) url.openConnection();
urlConn.setDoInput(true);
urlConn.setDoOutput(true);
urlConn.setUseCaches(false);
urlConn.setRequestMethod("GET");
BufferedReader input = new BufferedReader(new InputStreamReader(urlConn.getInputStream()));
String line;
StringBuilder sb = new StringBuilder();
while ((line = input.readLine()) != null) {
    sb.append(line);
}
input.close();
print(sb.toString().replaceAll("\n", "<br/>"));
```

Windows



Configuration below is based on setting proxy configuration parameters in the JVM environment. There's another option what is configuring proxy at OS level, consider looking for this information at Microsoft technet website.

Step one

Check starting and stopping OpenKM from the command line.

- First stop OpenKM service.
- Edit `$TOMCAT_HOME/bin/seten.bat` file and add the lines:



The <http://proxy.url.com> value must be your own url host of the proxy.

The "user" value must be the user for getting connection to the proxy. If your **proxy does not require authentication you can remove the entire line** where are set these parameters.

The "password" values must be the password of the user for getting connection to the proxy. If your **proxy does**

not require authentication you can remove the entire line where are set these parameters.

```
set JAVA_OPTS=%JAVA_OPTS% -Dhttp.proxyHost=http://proxy.url.com -Dhttps.proxyHost=htt
set JAVA_OPTS=%JAVA_OPTS% -Dhttp.proxyUser=user -Dhttps.proxyUser=user
set JAVA_OPTS=%JAVA_OPTS% -Dhttp.proxyPassword=password -Dhttps.proxyPassword=password
```

At the end your `setenv.bat` should looking something like that:

```
JAVA_OPTS="$JAVA_OPTS -Xms256m -Xmx2048m -XX:+UseConcMarkSweepGC -Djava.awt.headless=
JAVA_OPTS="$JAVA_OPTS -Dfile.encoding=UTF-8 -Dsun.jnu.encoding=UTF-8 -Dmail.mime.igno
set JAVA_OPTS=%JAVA_OPTS% -Dhttp.proxyHost=http://proxy.url.com -Dhttps.proxyHost=htt
set JAVA_OPTS=%JAVA_OPTS% -Dhttp.proxyUser=user -Dhttps.proxyUser=user
set JAVA_OPTS=%JAVA_OPTS% -Dhttp.proxyPassword=password -Dhttps.proxyPassword=password
CATALINA_PID=$CATALINA_HOME/catalina.pid
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$CATALINA_HOME/lib/sigar
```



If you have issues resolving DNS you can also try with this configuration what is trying to use Google DNS:

```
set JAVA_OPTS=%JAVA_OPTS% -Dsun.net.spi.nameservice.nameservers=8.8.8.8
set JAVA_OPTS=%JAVA_OPTS% -Dsun.net.spi.nameservice.provider.1=dns,sun
```

- **Start OpenKM from the command line.**

```
c:\tomcat-8.5.24\bin> catalina.bat run
```

- Go to **Administration** > **Tools** > [Scripting](#) and execute the script below (should be shown google html code).

```
import java.net.URL;
import java.net.HttpURLConnection;
import java.io.BufferedReader;

URL url = new URL("http://www.google.com");
HttpURLConnection urlConn = (HttpURLConnection) url.openConnection();
urlConn.setDoInput(true);
urlConn.setDoOutput(true);
urlConn.setUseCaches(false);
urlConn.setRequestMethod("GET");
BufferedReader input = new BufferedReader(new InputStreamReader(urlConn.getInputStream()));
String line;
StringBuilder sb = new StringBuilder();
while ((line = input.readLine()) != null) {
    sb.append(line);
}
input.close();
print(sb.toString().replaceAll("\n", "<br/>"));
```

Step two

If all has gone right in step one, now you can configure parameters into the OpenKM service.

- **Stop OpenKM.**

- Go to **\$TOMCAT_HOME/bin** and execute the next command line:



The command lines below does not registering the parameters for DNS resolving. If you need it, simply add them.

```
c:\tomcat-8.5.24\bin> OpenKM //US//OpenKM ++JvmOptions="-Dhttp.proxyHost=http://proxy.url.com"
c:\tomcat-8.5.24\bin> OpenKM //US//OpenKM ++JvmOptions="-Dhttps.proxyHost=http://proxy.url.com"
c:\tomcat-8.5.24\bin> OpenKM //US//OpenKM ++JvmOptions="-Dhttp.proxyUser=user"
c:\tomcat-8.5.24\bin> OpenKM //US//OpenKM ++JvmOptions="-Dhttps.proxyUser=user"
c:\tomcat-8.5.24\bin> OpenKM //US//OpenKM ++JvmOptions="-Dhttp.proxyPassword=password"
c:\tomcat-8.5.24\bin> OpenKM //US//OpenKM ++JvmOptions="-Dhttps.proxyPassword=password"
```

- **Start OpenKM from services.**
- Go to **Administration > Tools > [Scripting](#)** and execute the script below (should be shown google html code).

```
import java.net.URL;
import java.net.HttpURLConnection;
import java.io.BufferedReader;

URL url = new URL("http://www.google.com");
HttpURLConnection urlConn = (HttpURLConnection) url.openConnection();
urlConn.setDoInput(true);
urlConn.setDoOutput(true);
urlConn.setUseCaches(false);
urlConn.setRequestMethod("GET");
BufferedReader input = new BufferedReader(new InputStreamReader(urlConn.getInputStream()));
String line;
StringBuilder sb = new StringBuilder();
while ((line = input.readLine()) != null) {
    sb.append(line);
}
input.close();
print(sb.toString().replaceAll("\n", "<br/>"));
```

Change repository home

By default the repository is located at "\$TOMCAT_HOME/repository" but you can change to other location.



This action must be done with application stopped.

Understanding the contents of repository folders

Into "\$TOMCAT_HOME/repository" there are three folder "index", "cache" and "datastore".

Folder	Backup	Description
index	No	<p>The folder which contains the Lucene search engine indexes. The content of that folder can be rebuild.</p> <div> <p>Although is not mandatory backup this folder, we recommend doing it.</p> </div>
cache	No	<p>The folder which contains the application cached files. For example, for previewing documents, sometimes is needed to transform the original document to other format. The conversion consumes a lot of hardware resources, for this reason the application stores these converted files to be reused later. The content of that folder can be cleaned without major problems.</p> <div> <p>Although is not mandatory backup this folder, we recommend doing it.</p> </div>
datastore	Yes	<p>The folder which contains all binary data of the documents.</p> <div> <p>Is mandatory mandatory backup this folder otherwise the binary data of the documents could be lost on a server hardware fail.</p> </div>

Move entire repository location

- Stop the application
- Move the "\$TOMCAT_HOME/repository" to other location.
- Edit the "\$TOMCAT_HOME/OpenKM.cfg" file and add the property "**repository.home**".
- Start the application

Sample**Windows**

```
repository.home=C:/okmrepo
```

Linux

```
repository.home=/mnt/okmrepo
```

Move repository folders individually

Table of properties:

Folder	Property
index	<code>hibernate.search.index.home</code>
cache	<code>repository.cache.home</code>
datastore	<code>repository.datastore.home</code>



These parameters can not be used in combination with the parameter "**repository.home**".

- Stop the application
- Move the "**\$TOMCAT_HOME/repository/{folder}**" to other location.
- Edit the "**\$TOMCAT_HOME/OpenKM.cfg**" file and add the some property.
- Start the application

Sample**Windows**

```
hibernate.search.index.home=C:/okmrepo/index
repository.datastore.home=D:/okm/datastore
repository.cache.home=D:/okm/cache
```

Linux

```
hibernate.search.index.home=/mnt/okmrepo/index
repository.datastore.home=/mnt/okm/datastore
repository.cache.home=/mnt/okm/cache
```

Troubleshooting

Windows mapped network disk is not working



We encourage do not use networking (NAS or Samba) for storing repository. Because network issues will affect the repository we consider as a dangerous configuration.

*Seems there're restrictions using network unit (mapped with "**net use**" command line) and application running from services.*

To solve it is necessary to create a link between UNC and the file system.

```
MKLINK /D C:\okm-repository \\192.168.1.36\Documents\okm-repository
```



In most cases will be necessary to use a standard user for running OpenKM service either the standard configured. Take a look at OpenKM windows service configuration to change the default "Local system user" to your own.

Change default role names

By default application comes with two connection roles named "**ROLE_USER**" and "**ROLE_ADMIN**". Any OpenKM user must have assigned one of these roles to login.

Change the default roles



Change roles usually is not an easy task, we recommend have a backup before start working on it.

Is a good practice, first test this kind of changes on a development environment before trying on your production.

To change the default roles must be done three actions:

1. **Change the roles name** at administration.
2. **Modify the contents of the file** `applicationContext.xml` into the `"$TOMCAT/webapp/OpenKM.war"`.
3. **Update existing database roles** name "**ROLE_USER**" and "**ROLE_ADMIN**" to new names.



You can modify the file "`applicationContext.xml`" into the `"$TOMCAT_HOME/webapps/OpenKM.war"` or modify the exploded war file at `"$TOMCAT_HOME/webapp/OpenKM/WEB-INF/applicationContext.xml"`.

Take in mind each time the "`OpenKM.war`" file is modified the contents of `"$TOMCAT_HOME/webapp/OpenKM"` folder will be completely cleaned.



Sections into the "`appContext.xml`" to be modified:

Mandatory:

```
<security:intercept-url pattern="/admin/**" access="ROLE_ADMIN" />
```

Optional:

```
<!-- Remove prefix to be able of use custom roles -->
<beans:bean id="roleVoter" class="org.springframework.security.access.vote.RoleVoter">
  <beans:property name="rolePrefix" value="ROLE_" />
</beans:bean>
```

- Go to **Administration** > [Configuration parameters](#).
- Change the values of parameters "**default.admin.role**" and "**default.user.role**". Also can be done from Database query, see the next SQL script.
- Go to **Administration** > **Utilities** > [Database query](#) and execute the role name update sql:

```
-- Optionally the configuration parameters can be set from SQL
UPDATE OKM_CONFIG SET CFG_VALUE='NEW_ROLE_USER_NAME' WHERE CFG_KEY='default.user.role'
UPDATE OKM_CONFIG SET CFG_VALUE='NEW_ROLE_ADMIN_NAME' WHERE CFG_KEY='default.admin.role'
```

```
-- Update security table
UPDATE OKM_NODE_ROLE_PERMISSION SET NRP_ROLE='NEW_ROLE_USER_NAME' WHERE NRP_ROLE='ROLE_USER';
UPDATE OKM_NODE_ROLE_PERMISSION SET NRP_ROLE='NEW_ROLE_ADMIN_NAME' WHERE NRP_ROLE='ROLE_ADMIN';
-- Insert new roles
INSERT INTO OKM_ROLE (ROL_ID, ROL_ACTIVE) VALUES ('NEW_ROLE_USER_NAME', 'T');
INSERT INTO OKM_ROLE (ROL_ID, ROL_ACTIVE) VALUES ('NEW_ROLE_ADMIN_NAME', 'T');
-- Update user roles
UPDATE OKM_USER_ROLE SET UR_ROLE='NEW_ROLE_USER_NAME' WHERE UR_ROLE='ROLE_USER';
UPDATE OKM_USER_ROLE SET UR_ROLE='NEW_ROLE_ADMIN_NAME' WHERE UR_ROLE='ROLE_ADMIN';
-- Remove older roles
DELETE FROM OKM_ROLE WHERE ROL_ID IN ('ROLE_USER', 'ROLE_ADMIN');
```

- Stop the application.
- Modify the contents of the file "**applicationContext.xml**".
- Start the application.

Sample of applicationContext.xml file



The "ROLE_USER" has been modified to "OPENKM_ROLE_USER".

The "ROLE_ADMIN" has been modified to "OPENKM_ROLE_ADMIN".



The contents of the "applicationContext.xml" can be modified, take only as a reference.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns:beans="http://www.springframework.org/schema/beans"
  xmlns:security="http://www.springframework.org/schema/security"
  xmlns:context="http://www.springframework.org/schema/context"
  xmlns:jee="http://www.springframework.org/schema/jee"
  xmlns:jaxws="http://cxf.apache.org/jaxws"
  xmlns:jaxrs="http://cxf.apache.org/jaxrs"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/security
    http://www.springframework.org/schema/security/spring-security.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context.xsd
    http://www.springframework.org/schema/jee
    http://www.springframework.org/schema/jee/spring-jee.xsd
    http://cxf.apache.org/jaxws
    http://cxf.apache.org/schemas/jaxws.xsd
    http://cxf.apache.org/jaxrs
    http://cxf.apache.org/schemas/jaxrs.xsd">

  <context:component-scan base-package="com.openkm"/>

  <!-- <task:annotation-driven/> -->
  <!-- Tasks configuration moved to $CATALINA_HOME/OpenKM.xml -->

  <!-- Apache CXF Web Services -->
  <beans:import resource="classpath:META-INF/cxf/cxf.xml"/>
  <beans:import resource="classpath:META-INF/cxf/cxf-servlet.xml"/>

  <!--
  <beans:bean class="org.springframework.beans.factory.config.MethodInvokingFactoryBean"
    <beans:property name="targetClass" value="org.springframework.security.core.con
```

```

        <beans:property name="targetMethod" value="setStrategyName" />
        <beans:property name="arguments" value="_INHERITABLETHREADLOCAL" />
    </beans:bean>
    -->

    <beans:bean id="WSS4JInInterceptor" class="org.apache.cxf.ws.security.wss4j.WSS4JIn"
        <beans:constructor-arg>
            <beans:map>
                <beans:entry key="action" value="UsernameToken"/>
                <beans:entry key="passwordType" value="PasswordText"/>
                <beans:entry key="passwordCallbackClass" value="com.openkm.spring.ClientPassw
            </beans:map>
        </beans:constructor-arg>
    </beans:bean>

    <!-- SOAP -->
    <jaxws:endpoint id="authService" implementor="com.openkm.ws.endpoint.AuthService" a
    <jaxws:endpoint id="bookmarkService" implementor="com.openkm.ws.endpoint.BookmarkSe
    <jaxws:endpoint id="documentService" implementor="com.openkm.ws.endpoint.DocumentSe
    <jaxws:endpoint id="folderService" implementor="com.openkm.ws.endpoint.FolderServic
    <jaxws:endpoint id="mailService" implementor="com.openkm.ws.endpoint.MailService" a
    <jaxws:endpoint id="noteService" implementor="com.openkm.ws.endpoint.NoteService" a
    <jaxws:endpoint id="notificationService" implementor="com.openkm.ws.endpoint.Notifi
    <jaxws:endpoint id="propertyGroupService" implementor="com.openkm.ws.endpoint.Prope
    <jaxws:endpoint id="propertyService" implementor="com.openkm.ws.endpoint.PropertySe
    <jaxws:endpoint id="repositoryService" implementor="com.openkm.ws.endpoint.Reposito
    <jaxws:endpoint id="searchService" implementor="com.openkm.ws.endpoint.SearchServic
    <jaxws:endpoint id="dashboardService" implementor="com.openkm.ws.endpoint.Dashboard
    <jaxws:endpoint id="workflowService" implementor="com.openkm.ws.endpoint.WorkflowSe
    <jaxws:endpoint id="testService" implementor="com.openkm.ws.endpoint.TestService" a
    <!--
        <jaxws:inInterceptors>
            <beans:ref bean="WSS4JInInterceptor"/>
        </jaxws:inInterceptors>
    -->
</jaxws:endpoint>

    <!-- OpenCMIS -->
    <jaxws:endpoint id="cmisNavigationService" implementor="org.apache.chemistry.opencm
    <jaxws:endpoint id="cmisPolicyService" implementor="org.apache.chemistry.opencmis.s
    <jaxws:endpoint id="cmisDiscoveryService" implementor="org.apache.chemistry.opencmi
    <jaxws:endpoint id="cmisMultiFilingService" implementor="org.apache.chemistry.openc
    <jaxws:endpoint id="cmisRepositoryService" implementor="org.apache.chemistry.opencm
    <jaxws:endpoint id="cmisRelationshipService" implementor="org.apache.chemistry.open
    <jaxws:endpoint id="cmisVersioningService" implementor="org.apache.chemistry.opencm
    <jaxws:endpoint id="cmisObjectService" implementor="org.apache.chemistry.opencmis.s
    <jaxws:endpoint id="cmisAclService" implementor="org.apache.chemistry.opencmis.serv

    <!-- REST -->
    <jaxrs:server id="restAuth" address="/rest/auth">
        <jaxrs:serviceBeans>
            <beans:bean class="com.openkm.rest.endpoint.AuthService"/>
        </jaxrs:serviceBeans>
        <jaxrs:providers>
            <bean class="org.apache.cxf.jaxrs.provider.json.JSONProvider">
                <property name="dropRootElement" value="true"/>
                <property name="supportUnwrapped" value="true"/>
            </bean>
        </jaxrs:providers>
    </jaxrs:server>
    <jaxrs:server id="restDocument" address="/rest/document">
        <jaxrs:serviceBeans>
            <beans:bean class="com.openkm.rest.endpoint.DocumentService"/>
        </jaxrs:serviceBeans>
        <jaxrs:providers>
            <bean class="org.apache.cxf.jaxrs.provider.json.JSONProvider">
                <property name="dropRootElement" value="true"/>
            </bean>
        </jaxrs:providers>
    </jaxrs:server>

```

```
<property name="supportUnwrapped" value="true"/>
</bean>
</jaxrs:providers>
</jaxrs:server>
<jaxrs:server id="restFolder" address="/rest/folder">
  <jaxrs:serviceBeans>
    <beans:bean class="com.openkm.rest.endpoint.FolderService"/>
  </jaxrs:serviceBeans>
  <jaxrs:providers>
    <bean class="org.apache.cxf.jaxrs.provider.json.JSONProvider">
      <property name="dropRootElement" value="true"/>
      <property name="supportUnwrapped" value="true"/>
    </bean>
  </jaxrs:providers>
</jaxrs:server>
<jaxrs:server id="restMail" address="/rest/mail">
  <jaxrs:serviceBeans>
    <beans:bean class="com.openkm.rest.endpoint.MailService"/>
  </jaxrs:serviceBeans>
  <jaxrs:providers>
    <bean class="org.apache.cxf.jaxrs.provider.json.JSONProvider">
      <property name="dropRootElement" value="true"/>
      <property name="supportUnwrapped" value="true"/>
    </bean>
  </jaxrs:providers>
</jaxrs:server>
<jaxrs:server id="restNote" address="/rest/note">
  <jaxrs:serviceBeans>
    <beans:bean class="com.openkm.rest.endpoint.NoteService"/>
  </jaxrs:serviceBeans>
  <jaxrs:providers>
    <bean class="org.apache.cxf.jaxrs.provider.json.JSONProvider">
      <property name="dropRootElement" value="true"/>
      <property name="supportUnwrapped" value="true"/>
    </bean>
  </jaxrs:providers>
</jaxrs:server>
<jaxrs:server id="restPropertyGroup" address="/rest/propertyGroup">
  <jaxrs:serviceBeans>
    <beans:bean class="com.openkm.rest.endpoint.PropertyGroupService"/>
  </jaxrs:serviceBeans>
  <jaxrs:providers>
    <bean class="org.apache.cxf.jaxrs.provider.json.JSONProvider">
      <property name="dropRootElement" value="true"/>
      <property name="supportUnwrapped" value="true"/>
    </bean>
  </jaxrs:providers>
</jaxrs:server>
<jaxrs:server id="restSearch" address="/rest/search">
  <jaxrs:serviceBeans>
    <beans:bean class="com.openkm.rest.endpoint.SearchService"/>
  </jaxrs:serviceBeans>
  <jaxrs:providers>
    <bean class="org.apache.cxf.jaxrs.provider.json.JSONProvider">
      <property name="dropRootElement" value="true"/>
      <property name="supportUnwrapped" value="true"/>
    </bean>
  </jaxrs:providers>
</jaxrs:server>
<jaxrs:server id="restRepository" address="/rest/repository">
  <jaxrs:serviceBeans>
    <beans:bean class="com.openkm.rest.endpoint.RepositoryService"/>
  </jaxrs:serviceBeans>
  <jaxrs:providers>
    <bean class="org.apache.cxf.jaxrs.provider.json.JSONProvider">
      <property name="dropRootElement" value="true"/>
      <property name="supportUnwrapped" value="true"/>
    </bean>
  </jaxrs:providers>
</jaxrs:server>
```

```

        </bean>
    </jaxrs:providers>
</jaxrs:server>
<jaxrs:server id="restProperty" address="/rest/property">
    <jaxrs:serviceBeans>
        <beans:bean class="com.openkm.rest.endpoint.PropertyService"/>
    </jaxrs:serviceBeans>
    <jaxrs:providers>
        <bean class="org.apache.cxf.jaxrs.provider.json.JSONProvider">
            <property name="dropRootElement" value="true"/>
            <property name="supportUnwrapped" value="true"/>
        </bean>
    </jaxrs:providers>
</jaxrs:server>
<jaxrs:server id="restTest" address="/rest/test">
    <jaxrs:serviceBeans>
        <beans:bean class="com.openkm.rest.endpoint.TestService"/>
    </jaxrs:serviceBeans>
    <jaxrs:providers>
        <bean class="org.apache.cxf.jaxrs.provider.json.JSONProvider">
            <property name="dropRootElement" value="true"/>
            <property name="supportUnwrapped" value="true"/>
        </bean>
    </jaxrs:providers>
</jaxrs:server>

<security:global-method-security secured-annotations="enabled"/>

<!-- Remove prefix to be able to use custom roles -->
<beans:bean id="roleVoter" class="org.springframework.security.access.vote.RoleVote
    <beans:property name="rolePrefix" value="OPENKM_ROLE_" />
</beans:bean>

<!-- OpenCMIS -->
<beans:bean id="CmisLifecycleBean" class="com.openkm.cmis.CmisLifecycleBean">
    <beans:property name="cmisServiceFactory" ref="CmisServiceFactory"/>
</beans:bean>
<beans:bean id="CmisServiceFactory" class="com.openkm.cmis.CmisServiceFactory"/>

<!-- Status -->
<security:http pattern="/Status" create-session="stateless">
    <security:intercept-url pattern="/*" access="IS_AUTHENTICATED_FULLY"/>
    <security:http-basic/>
</security:http>

<!-- Download -->
<security:http pattern="/Download" create-session="stateless">
    <security:intercept-url pattern="/*" access="IS_AUTHENTICATED_FULLY"/>
    <security:http-basic/>
</security:http>

<!-- Workflow deploy -->
<security:http pattern="/workflow-register" create-session="stateless">
    <security:intercept-url pattern="/*" access="IS_AUTHENTICATED_FULLY"/>
    <security:http-basic/>
</security:http>

<!-- WebDAV using Basic authentication -->
<security:http pattern="/webdav/*" create-session="stateless">
    <security:intercept-url pattern="/*" access="IS_AUTHENTICATED_FULLY"/>
    <security:http-basic/>
</security:http>

<!-- Syndication using Basic authentication -->
<security:http pattern="/feed/*" create-session="stateless">
    <security:intercept-url pattern="/*" access="IS_AUTHENTICATED_FULLY"/>
    <security:http-basic/>

```

```

</security:http>

<!-- OpenCMIS (Browser) using Basic authentication -->
<security:http pattern="/cmis/browser/**" create-session="stateless">
  <security:intercept-url pattern="/**" access="IS_AUTHENTICATED_FULLY"/>
  <security:http-basic/>
</security:http>

<!-- OpenCMIS (AtomPub) using Basic authentication -->
<security:http pattern="/cmis/atom/**" create-session="stateless">
  <security:intercept-url pattern="/**" access="IS_AUTHENTICATED_FULLY"/>
  <security:http-basic/>
</security:http>

<!-- OpenCMIS (AtomPub) using Basic authentication -->
<security:http pattern="/cmis/atom11/**" create-session="stateless">
  <security:intercept-url pattern="/**" access="IS_AUTHENTICATED_FULLY"/>
  <security:http-basic/>
</security:http>

<!-- REST -->
<security:http pattern="/services/rest/**" create-session="stateless">
  <security:intercept-url pattern="/**" access="IS_AUTHENTICATED_FULLY"/>
  <security:http-basic/>
</security:http>

<!-- Additional filter chain for normal users, matching all other requests -->
<!-- http://info.michael-simons.eu/2011/01/28/disable-jsessionid-path-parameter-in-
<security:http access-decision-manager-ref="accessDecisionManager" access-denied-pa

  <!-- GWT -->
  <security:intercept-url pattern="/frontend/**" access="IS_AUTHENTICATED_FULLY"/>

  <!-- JSPs -->
  <security:intercept-url pattern="/login.jsp" access="IS_AUTHENTICATED_ANONYMOUSLY"
  <security:intercept-url pattern="/admin/**" access="OPENKM_ROLE_ADMIN"/>
  <security:intercept-url pattern="/mobile/**" access="IS_AUTHENTICATED_FULLY"/>

  <!-- Servlets -->
  <security:intercept-url pattern="/RepositoryStartup" access="IS_AUTHENTICATED_FUL
  <security:intercept-url pattern="/TextToSpeech" access="IS_AUTHENTICATED_FULLY"/>
  <security:intercept-url pattern="/HtmlPreview" access="IS_AUTHENTICATED_FULLY"/>
  <security:intercept-url pattern="/SyntaxHighlighter" access="IS_AUTHENTICATED_FUL
  <security:intercept-url pattern="/Test" access="IS_AUTHENTICATED_FULLY"/>

  <!-- Extensions -->
  <security:intercept-url pattern="/extension/ZohoFileUpload" access="IS_AUTHENTICA
  <security:intercept-url pattern="/extension/**" access="IS_AUTHENTICATED_FULLY"/>

  <!-- Login page -->
  <security:form-login login-page="/login.jsp" authentication-failure-url="/login.j

</security:http>

<!-- Needed for changing default role prefix -->
<beans:bean id="accessDecisionManager" class="org.springframework.security.access.v
  <beans:property name="decisionVoters">
    <beans:list>
      <beans:bean class="org.springframework.security.web.access.expression.WebExpr
      <beans:ref bean="roleVoter"/>
      <beans:bean class="org.springframework.security.access.vote.AuthenticatedVote
    </beans:list>
  </beans:property>
</beans:bean>

<!-- Security access logger -->
<beans:bean id="loggerListener" class="com.openkm.spring.LoggerListener"/>

```

```
<jee:jndi-lookup id="dataSource" jndi-name="jdbc/OpenKMDS" resource-ref="true"/>
<!-- Security configuration moved to $CATALINA_HOME/OpenKM.xml -->
<!-- WINFIX
<security:authentication-manager alias="authenticationManager">
  <security:authentication-provider>
    <security:password-encoder hash="md5"/>
    <security:jdbc-user-service
      data-source-ref="dataSource"
      users-by-username-query="select usr_id, usr_password, 1 from OKM_USER w.
      authorities-by-username-query="select ur_user, ur_role from OKM_USER_RO.
    </security:authentication-provider>
  </security:authentication-manager>
  WINFIX -->
</beans:beans>
```

Tools

Download the [changeRolesTools.zip](#).

It includes:

- `change_roles.sh` script to modify the `applicationContext.xml` file.
- `path` folder to patch "**OpenKM.war**" file.



To patch "**OpenKM.war**" file, copy the file into `patch` folder and then execute the `patch.sh` script (or `patch.bat` file on Windows).

Enable extensions



After the extensions have been registered in OpenKM will not be visible for the users until you enable them from the [Profile General tab](#).

By default OpenKM comes with the extensions disabled.

Types of extensions:

- Do not require additional software configuration.
- Need additional software configuration.
- Sample extensions for development.

Available extensions are registered in the `OKM_EXTENSION` table. To enable any of them, it's only necessary to insert it in this table.

Extensions that do not require additional software configuration:

- Go to **Administration > Database query**. At bottom right select option **JDBC**, paste into the box the SQL inserts of the extensions you want to enable. Finally click at bottom right the button **execute**.



The **Stapling** extension is deprecated and has been superseded by **Relation**.

```
INSERT INTO OKM_EXTENSION (EXT_UUID, EXT_NAME) VALUES ('25af39c0-580f-431c-8852-0b643
INSERT INTO OKM_EXTENSION (EXT_UUID, EXT_NAME) VALUES ('a1925a00-ef41-11df-98cf-08002
INSERT INTO OKM_EXTENSION (EXT_UUID, EXT_NAME) VALUES ('88ca0d10-39e2-11e0-9207-08002
INSERT INTO OKM_EXTENSION (EXT_UUID, EXT_NAME) VALUES ('522e7720-5c54-11e0-80e3-08002
INSERT INTO OKM_EXTENSION (EXT_UUID, EXT_NAME) VALUES ('a7b5a3c0-4b2f-11e0-b8af-08002
INSERT INTO OKM_EXTENSION (EXT_UUID, EXT_NAME) VALUES ('bb3feb20-570b-11e0-b8af-08002
INSERT INTO OKM_EXTENSION (EXT_UUID, EXT_NAME) VALUES ('fa7f4556-3249-4268-88e0-0dd78.
INSERT INTO OKM_EXTENSION (EXT_UUID, EXT_NAME) VALUES ('c60082c2-7d4c-4750-901b-a817f.
INSERT INTO OKM_EXTENSION (EXT_UUID, EXT_NAME) VALUES ('87f250d5-526d-4d8a-96ee-1e5be
INSERT INTO OKM_EXTENSION (EXT_UUID, EXT_NAME) VALUES ('c20c69a8-7d6b-4539-9e6b-6f2c1
INSERT INTO OKM_EXTENSION (EXT_UUID, EXT_NAME) VALUES ('101fa1e6-4bf6-4e39-9124-88f44.
```

Deprecated extensions:

```
INSERT INTO OKM_EXTENSION (EXT_UUID, EXT_NAME) VALUES ('df5eb783-fb06-4b4b-bc89-4fdaa.
```

Sample extensions for development:

There're some sample extension for development testing purpose.

- Go to **Administration > Database query**. At bottom right select option **JDBC**, paste into the box the SQL inserts of the extensions you want to enable. Finally click at bottom right the button **execute**.



```
INSERT INTO OKM_EXTENSION (EXT_UUID,EXT_NAME) VALUES ('9f84b330-d096-11df-bd3b-080020000000')
INSERT INTO OKM_EXTENSION (EXT_UUID,EXT_NAME) VALUES ('44f94470-d097-11df-bd3b-080020000000')
INSERT INTO OKM_EXTENSION (EXT_UUID,EXT_NAME) VALUES ('d95e01a0-d097-11df-bd3b-080020000000')
INSERT INTO OKM_EXTENSION (EXT_UUID,EXT_NAME) VALUES ('d9dab640-d098-11df-bd3b-080020000000')
```

Enable WebDAV

Once WebDAV is enabled, you can access the OpenKM repository with this URL:

<http://your-domain.com:8080/OpenKM/webdav>

Go to **Administration > Configuration parameters**:

Field / Property	Type	Description
system.webdav.server	Boolean	<p>Enable WebDAV service. By default it is disabled.</p> <p>true</p>
system.webdav.fix	Boolean	<p>Character ":" may cause problems with some WebDAV clients, Enabling this parameter will replace in the URL the character ":" by "_". By default is disabled.</p> <div>  <p>It's recommended enabling this parameter to prevent compatibility problems with some WebDAV clients.</p> <p>A path like <code>/okm:root/myfile.txt</code> will be translated to <code>/okm_root/myfile.txt</code></p> </div> <p>true</p>



OpenKM WebDAV is based on excellent Open Source [Milton WebDAV API for Java](#). The [Milton project](#) maintains a detailed [WebDAV client compatibility list](#) that describes a "Recipe for broad client compatibility."

Linux

From Linux you can use:

- <http://dav.sourceforge.net/>
- <http://savannah.nongnu.org/projects/davfs2>

For example, in Ubuntu or any Debian based Linux distro you can install this utility this way:

```
$ sudo aptitude install davfs2
```

Make a mount point:

```
$ sudo mkdir ~/okmdav
```

And mount it:

```
$ sudo mount.davfs "http://your-domain.com:8080/OpenKM/webdav" ~/okmdav
```

To ease the process, add this entry to your `/etc/fstab`:

```
http://your-domain.com:8080/OpenKM/webdav /home/username/webdav davfs noauto,user,rw
```

And then in the user's home directory, look for `./davfs2/davfs2.conf`. There you'll find a `if_match_bug 1` line. Uncomment it.

See also:

- [Ubuntu: Mounting remote filesystem using davfs2 \(FUSE\)](#)

Windows

Install a WebDAV client for Windows.



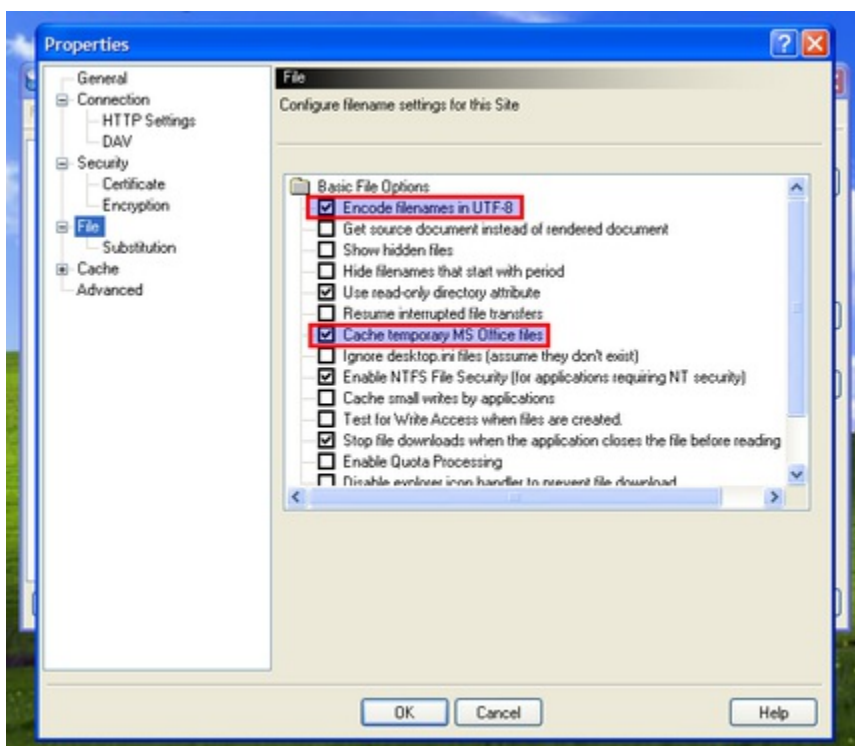
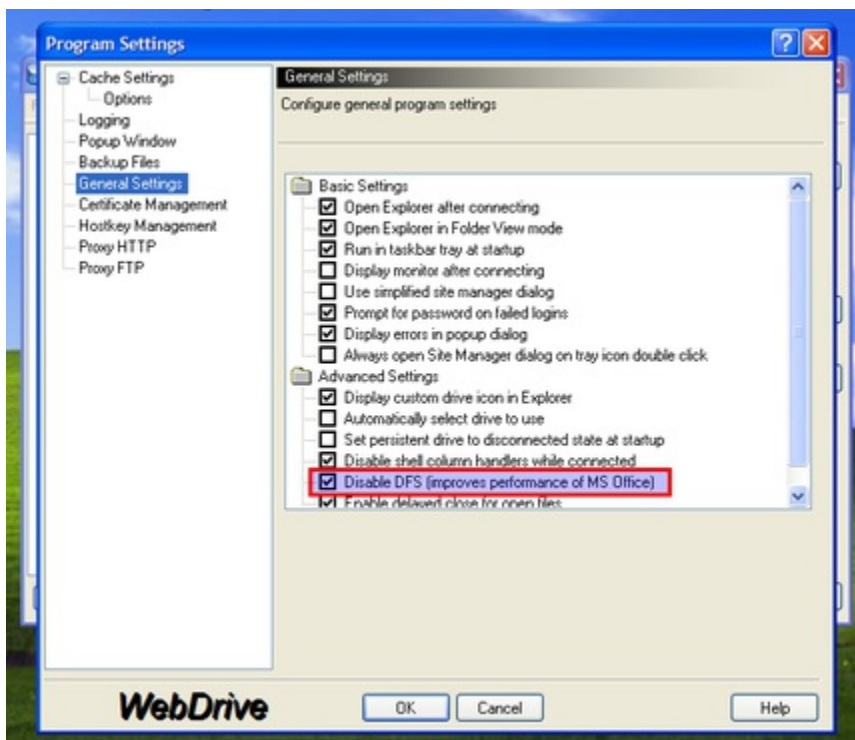
We don't recommend using the default Windows WebDAV support because is buggy.

Here you have several options:

- <http://www.netdrive.net/> (Free for home use but does not handle document versioning properly)
- <http://www.webdrive.com/> (The recommended option and will create a new document version when you save the modifications)
- [WebDAV Clients on Windows XP](#)

WebDrive

Probably this is the best choice if you are under Windows, because of its stability and ease of configuration. WebDrive 10 (Version 10.00, Build 2521) has been roughly tested with OpenKM to enhance compatibility effort. This is the supported configuration to work with OpenKM WebDAV connector:



Windows Web Folders

Perform the following steps in Internet Explorer:

- Select File -> Open.

- Fill in URL like <http://your-domain.com:8080/OpenKM/webdav>
- Check "Open as Web Folder".
- Click OK.



There are some well-known issues. Please consult the following articles in case of any problem

- *Be aware that there are multiple versions of WebDAV Microsoft libraries (and different type of bugs). To avoid some frustration if these steps don't work for you [Update Windows XP for Web Folders](#), more information at [Web Folder Client \(MSDAIPP.DLL\) Versions and Issues](#).*
- *Windows Vista and Windows 7 both restrict access to WebDAV servers that use Basic HTTP authentication on non-SSL connections. This restriction can be solved by changing a registry key. Read more on. However, SSL connections do bring improved security. [This page from Greenbytes](#) is an excellent resource to analyse & fix problems with your Windows WebDAV client.*

Windows XP's "My Network Places"

Windows XP supports 'Network Places' in the standard File dialog:

- Open networking.
- Connect as network resource.
- Write url <http://your-domain.com:8080/OpenKM/webdav>
- Set user and password.
- Open the Registry Editor utility (regedit.exe).
- Go to **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WebClient\Parameters**, look for **TypeUseBasicAuth** and replace the default value "1" by "2" allowing BasicAuth through NON SSL connections.
- Map your drive using this address: `\\your-domain.com:8080/OpenKM/repository/default/okm_root/`

More information at our [OpenKM Forum](#).

Other tools

- [Raiddrive](#)

Mac

Install any WebDAV client for Mac, here you have some options:

- <http://cyberduck.ch/>
- <http://www.webdrive.com/> (There is a version for Macintosh OS/X)

Mac OS X finder.

- Select in the Finder "**Go**" and "**Connect to Server...**".
- Fill in the WebDAV URL <http://your-domain:8080/OpenKM/webdav>
- Enter a username and password.
- And the document repository is accessible!

Troubleshooting

Slow WebDAV performance on Windows

If you have a slow webdav connection in windows 7 please follow the instructions in [Fix Slow WebDAV Performance in Windows 7](#). It turns out to be a windows issue, not OpenKM.

Furthermore in Windows 7/Vista you need to add your certificate (if its not issued by a verified CA) into windows if you plan to use Webdav over SSL (and, like me, do not compensate with digest auth):

1. Get the certificate from the server administrator in CER format or download it from Firefox.
2. Start windows Certificate Manager: in command line type **certmgr.msc**
3. Select on the left **Trusted Root Certification Authorities**.
4. In the menu **Actions > All Tasks** select **Import** and then **Next**.
5. Browse for the location of certificate file you saved earlier on your computer and click **Open**.
6. You will get a few more warning messages. Just say to go through with it until the certificate is accepted.

Then map webdav content as a network drive without any other software. The folder should be: <http://your-domain:8080/OpenKM/webdav>

To make sure that everything is ok with certificates, open the address using IE. If no warning pops up is shown, then is right.



This allows you only to read/view files. If you need write access over a network drive you need to disable File Locking (windows does not support webdav file lock).

For more information [Web Lockers Webdav For MS Windows](#)



For Windows XP there is a fix (KB907306).

Turn off file locking

If you need to write access over a network drive you need to disable File Locking (windows does not support webdav file lock).

Execute registry editor tool, in command line type **regedit**. Modify or create this configuration parameter:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WebClient\Parameters]
"SupportLocking"=dword:00000000
```

LDAP configuration

The LDAP integration is divided in two sections, login process and retrieving users and roles from LDAP server. It's usually better starting with [LDAP configuration parameters](#) which are used by OpenKM to show users and roles into administration and once this section is configured go to Spring Security configuration (`OpenKM.xml` changes).

Usually you only want to retrieve a subset of users and roles present in your LDAP, to be shown in the user interface or be able to login into OpenKM. If this is your case, please take a look at [LDAP best practices for filtering users and roles](#) .

OpenLDAP

- [OpenLDAP example with URL base at login](#)

Active Directory

- [Active Directory basic configuration](#)
- [Active Directory advanced configuration](#)
- [Active Directory example with referral enabled](#)
- [Active Directory example with login based on filtering users by roles](#) (suggested)

Uncategorised examples shared by users

- <https://www.nosam.com/node/8>, <http://pastebin.com/iQa4LhyY> (more information [OpenKM public forum](#))

Tools

- [Microsoft Active Directoy Explorer](#).
- [Apache Directory Studio](#).

OpenLDAP example with URL base at login

LDAP Structure

```
dc=com
  dc=some
    ou=organization
      cn=ROLE_ADMIN
        memberUid=okmAdmin
        memberUid=user1
        memberUid=user2
      cn=ROLE_USER
        memberUid=user3
        memberUid=user4
      ...
    ou=organization
      uid=user1
        mail=user@mail.com
        cn=User Name 1
      uid=user2
        mail=user2@mail.com
        cn=User Name 3
      uid=user3
        mail=user3@mail.com
        cn=User Name 3
      uid=user4
        mail=user4@mail.com
        cn=User Name 4
```

Valid roles:

- ***cn=ROLE_X,ou=roles,dc=some,dc=com***
- ***cn=ROLE_Y,ou=dept marketing,ou=roles,dc=some,dc=com***
- ***cn=ROLE_Z,ou=dept sales,ou=roles,dc=some,dc=com***

Invalid roles:

- ***cn=ROLE_INVALID,ou=dept,dc=some,dc=com*** (any distinguished name not included in ***ou=roles,dc=some,dc=com***)

Valid users:

- ***uid=USER_X,ou=organization,dc=some,dc=com***
- ***uid=USER_Y,ou=dept id,ou=organization,dc=some,dc=com***
- ***uid=USER_Z,ou=dept administrator,ou=organization,dc=some,dc=com***

Invalid users:

- ***uid=USER_INVALID,ou=house,dc=some,dc=com*** (any distinguished name not included in ***ou=organization,dc=some,dc=com***)

OpenKM.xml

- **Important** login (`ldap://192.168.0.13:389/dc=some,dc=com`) sets default filter base queries at `dc=some,dc=com` with is concatenated by default in all filter queries .
- Roles (groups) filter base is `ou=roles` (real distinguished name is `ou=roles,dc=some,dc=com`). Any valid roles should have it as parent. `ou=roles"/>` really points to `ou=roles,dc=some,dc=com"/>`
- Users filter base is `ou=organization` (real distinguished name is `ou=organization,dc=some,dc=com`). Any valid user should have it as parent. to
- User filter is `uid={0}`
- Finally take in consideration the value 1 at `memberUid={1}"/>`

```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns:beans="http://www.springframework.org/schema/beans"
  xmlns:security="http://www.springframework.org/schema/security"
  xmlns:task="http://www.springframework.org/schema/task"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/security
    http://www.springframework.org/schema/security/spring-security.xsd
    http://www.springframework.org/schema/task
    http://www.springframework.org/schema/task/spring-tasks.xsd">

  <security:authentication-manager alias="authenticationManager">
    <security:authentication-provider ref="ldapAuthProvider" />
  </security:authentication-manager>

  <beans:bean id="contextSource" class="org.springframework.security.ldap.DefaultSpringSecurityContextSource">
    <beans:constructor-arg value="ldap://192.168.0.13:389/dc=some,dc=com"/>
    <beans:property name="userDn" value="cn=Manager,dc=some,dc=com"/>
    <beans:property name="password" value="*****"/>
  </beans:bean>

  <beans:bean id="ldapAuthProvider" class="org.springframework.security.ldap.authentication.LdapAuthenticationProvider">
    <beans:constructor-arg>
      <beans:bean class="org.springframework.security.ldap.authentication.BindAuthenticationHandler">
        <beans:constructor-arg ref="contextSource"/>
        <beans:property name="userSearch" ref="userSearch"/>
      </beans:bean>
    </beans:constructor-arg>
    <beans:constructor-arg>
      <beans:bean class="org.springframework.security.ldap.userdetails.DefaultLdapAuthoritiesResolver">
        <beans:constructor-arg ref="contextSource"/>
        <beans:constructor-arg value="ou=roles"/>
        <beans:property name="groupSearchFilter" value="memberUid={1}"/>
        <beans:property name="groupRoleAttribute" value="cn"/>
        <beans:property name="searchSubtree" value="true" />
        <beans:property name="convertToUpperCase" value="true" />

        <beans:property name="rolePrefix" value="" />
      </beans:bean>
    </beans:constructor-arg>
  </beans:bean>

  <beans:bean id="userSearch" class="org.springframework.security.ldap.search.FilterBasedLdapUserSearch">
    <beans:constructor-arg index="0" value="ou=organization" />
    <beans:constructor-arg index="1" value="uid={0}" />
```

```

    <beans:constructor-arg index="2" ref="contextSource" />
    <beans:property name="searchSubtree" value="true" />
  </beans:bean>

</beans:beans>

```

Configuration parameters

Go to **Administration > Configuration parameters**:

Field / Property	Type	Description
principal.adapter	String	com.openkm.principal.LdapPrincipalAdapter
system.login.lowercase	String	false
principal.ldap.server	String	ldap://192.168.xxx.xxx:389
principal.ldap.security.principal	String	cn=Manager,dc=some,dc=com
principal.ldap.security.credentials	String	password
principal.ldap.referral	String	
principal.ldap.users.from.roles	Boolean	false
principal.ldap.user.attribute	String	uid
principal.ldap.user.search.base	List	ou=users,dc=some,dc=com
principal.ldap.user.search.filter	String	(objectClass=inetOrgPerson)
principal.ldap.username.attribute	String	cn
principal.ldap.username.search.base	String	ou=users,dc=some,dc=com
principal.ldap.username.search.filter	String	(uid={0})
principal.ldap.mail.attribute	String	mail
principal.ldap.mail.search.base	String	dc=some,dc=com

<i>principal.ldap.mail.search.filter</i>	<i>String</i>	<i>(uid={0})</i>
<i>principal.ldap.role.attribute</i>	<i>String</i>	<i>cn</i>
<i>principal.ldap.role.search.base</i>	<i>List</i>	<i>ou=roles,dc=some,dc=com</i>
<i>principal.ldap.role.search.filter</i>	<i>String</i>	<i>(objectClass=posixGroup)</i>
<i>principal.ldap.roles.by.user.attribute</i>	<i>String</i>	<i>cn</i>
<i>principal.ldap.roles.by.user.search.base</i>	<i>String</i>	<i>ou=roles,dc=some,dc=com</i>
<i>principal.ldap.roles.by.user.search.filter</i>	<i>String</i>	<i>(memberUid={0})</i>
<i>principal.ldap.users.by.role.attribute</i>	<i>String</i>	<i>memberUid</i>
<i>principal.ldap.users.by.role.search.base</i>	<i>String</i>	<i>ou=roles,dc=some,dc=com</i>
<i>principal.ldap.users.by.role.search.filter</i>	<i>String</i>	<i>(&(objectClass=posixGroup)(cn={0}))</i>

OpenLDAP example retrieving attributes by name

LDAP Structure

```
dc=com
  dc=some
    ou=roles
      cn=ROLE_ADMIN
        member=cn=okmAdmin,ou=users,dc=some,dc=com
        member=cn=user1,ou=users,dc=some,dc=com
        member=cn=user2,ou=users,dc=some,dc=com
      cn=ROLE_USER
        member=cn=user3,ou=users,dc=some,dc=com
      ...
    ou=users
      uid=user1
        mail=user@mail.com
        cn=User Name 1
        memberOf=cn=ROLE_ADMIN,ou=roles,dc=some,dc=com
      uid=user2
        mail=user2@mail.com
        cn=User Name 3
        memberOf=cn=ROLE_ADMIN,ou=roles,dc=some,dc=com
      uid=user3
        mail=user3@mail.com
        cn=User Name 3
        memberOf=cn=ROLE_USER,ou=roles,dc=some,dc=com
      uid=Manager
        mail=manager@mail.com
        cn=Manager
```

Valid roles:

- ***cn=ROLE_X,ou=roles,dc=some,dc=com***
- ***cn=ROLE_Y,ou=dept marketing,ou=roles,dc=some,dc=com***
- ***cn=ROLE_Z,ou=dept sales,ou=roles,dc=some,dc=com***

Invalid roles:

- ***cn=ROLE_INVALID,ou=dept,dc=some,dc=com*** (any distinguished name not included in *ou=roles,dc=some,dc=com*)

Valid users:

- ***uid=USER_X,ou=users,dc=some,dc=com***
- ***uid=USER_Y,ou=dept id,ou=users,dc=some,dc=com***
- ***uid=USER_Z,ou=dept administrator,ou=users,dc=some,dc=com***

Invalid users:

- ***uid=USER_INVALID,ou=house,dc=some,dc=com*** (any distinguished name not included in *ou=organization,dc=some,dc=com*)

OpenKM.xml

- **Important** login (`ldap://192.168.0.13:389/dc=some,dc=com`) sets default filter base queries at `dc=some,dc=com` with is concatenated by default in all filter queries .
- Roles (groups) filter base is `ou=roles` (real distinguished name is `ou=roles,dc=some,dc=com`). Any valid roles should have it as parent. `ou=roles"/>` really points to `ou=roles,dc=some,dc=com"/>`
- Users filter base is `ou=users` (real distinguished name is `ou=users,dc=some,dc=com`). Any valid user should have it as parent. to
- User filter is `uid={0}`
- Finally take in consideration the value 1 at `member={0}"/>`

```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns:beans="http://www.springframework.org/schema/beans"
  xmlns:security="http://www.springframework.org/schema/security"
  xmlns:task="http://www.springframework.org/schema/task"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/security/spring-security.xsd
    http://www.springframework.org/schema/task/spring-tasks.xsd">

  <security:authentication-manager alias="authenticationManager">
    <security:authentication-provider ref="ldapAuthProvider" />
  </security:authentication-manager>

  <beans:bean id="contextSource" class="org.springframework.security.ldap.DefaultSpringSecurityContextSource">
    <beans:constructor-arg value="ldaps://192.168.0.13:389/dc=some,dc=com"/>
    <beans:property name="userDn" value="cn=Manager,ou=users,dc=some,dc=com"/>
    <beans:property name="password" value="***"/>
  </beans:bean>

  <beans:bean id="ldapAuthProvider" class="org.springframework.security.ldap.authentication.LdapAuthenticationProvider">
    <beans:constructor-arg>
      <beans:bean class="org.springframework.security.ldap.authentication.BindAuthHandler">
        <beans:constructor-arg ref="contextSource"/>
        <beans:property name="userSearch" ref="userSearch"/>
      </beans:bean>
    </beans:constructor-arg>
    <beans:constructor-arg>
      <beans:bean class="org.springframework.security.ldap.userdetails.DefaultLdapUserDetailsContext">
        <beans:constructor-arg ref="contextSource"/>
        <beans:constructor-arg value="ou=roles"/>
        <beans:property name="groupSearchFilter" value="member={0}"/>
        <beans:property name="groupRoleAttribute" value="cn"/>
        <beans:property name="searchSubtree" value="true" />
        <beans:property name="convertToUpperCase" value="true" />
      </beans:bean>
    </beans:constructor-arg>
    <beans:property name="rolePrefix" value="" />
  </beans:bean>

  <beans:bean id="userSearch" class="org.springframework.security.ldap.search.FilterBasedLdapUserSearch">
    <beans:constructor-arg index="0" value="ou=users" />
    <beans:constructor-arg index="1" value="uid={0}" />
  </beans:bean>
</beans:beans>
```

```

    <beans:constructor-arg index="2" ref="contextSource" />
    <beans:property name="searchSubtree" value="true" />
  </beans:bean>

</beans:beans>

```

Configuration parameters

Go to **Administration > Configuration parameters**:

Field / Property	Type	Description
principal.adapter	String	com.openkm.principal.OpenLdapPrincipalAdapter
system.login.lowercase	String	false
principal.ldap.server	String	ldap://192.168.xxx.xxx:389
principal.ldap.security.principal	String	cn=Manager,ou=users,dc=some,dc=com
principal.ldap.security.credentials	String	password
principal.ldap.referral	String	
principal.ldap.users.from.roles	Boolean	false
principal.ldap.user.attribute	String	uid
principal.ldap.user.search.base	List	ou=users,dc=some,dc=com
principal.ldap.user.search.filter	String	(objectClass=inetOrgPerson)
principal.ldap.username.attribute	String	cn
principal.ldap.username.search.base	String	ou=users,dc=some,dc=com
principal.ldap.username.search.filter	String	(uid={0})
principal.ldap.mail.attribute	String	mail
principal.ldap.mail.search.base	String	dc=some,dc=com

<i>principal.ldap.mail.search.filter</i>	<i>String</i>	<code>(uid={0})</code>
<i>principal.ldap.role.attribute</i>	<i>String</i>	<code>cn</code>
<i>principal.ldap.role.search.base</i>	<i>List</i>	<code>ou=roles,dc=some,dc=com</code>
<i>principal.ldap.role.search.filter</i>	<i>String</i>	<code>(objectClass=groupOfNames)</code>
<i>principal.ldap.roles.by.user.attribute</i>	<i>String</i>	<code>memberOf</code>
<i>principal.ldap.roles.by.user.search.base</i>	<i>String</i>	<code>ou=users,dc=some,dc=com</code>
<i>principal.ldap.roles.by.user.search.filter</i>	<i>String</i>	<code>(&(objectClass=inetOrgPerson)(uid={0}))</code>
<i>principal.ldap.users.by.role.attribute</i>	<i>String</i>	<code>member</code>
<i>principal.ldap.users.by.role.search.base</i>	<i>String</i>	<code>ou=roles,dc=some,dc=com</code>
<i>principal.ldap.users.by.role.search.filter</i>	<i>String</i>	<code>(&(objectClass=groupOfNames)(cn={0}))</code>

Active Directory example with referral enabled

This example covers the scenario where more than one Active Directory domains work together, and it's needed the **following** parameter.

LDAP Structure

```
dc=com
  dc=company
    ou=OPENKM
      cn=ROLE_ADMIN
        member=okmAdmin
        member=user1
        member=user2
      cn=ROLE_USER
        member=user3
        member=user4
      cn=ROLE_XXXX
      cn=ROLE_YYYY
      ...
    ou=organization1
      sAMAccountName=okmAdmin
        memberOf=CN=ROLE_ADMIN,OU=OPENKM,DC=company,DC=com
        mail=okmAdmin@mail.com
        cn=OpenKM Administrator
      sAMAccountName=user1
        memberOf=CN=ROLE_ADMIN,OU=OPENKM,DC=company,DC=com
        mail=user1@mail.com
        cn=User Name 1
      sAMAccountName=user2
        memberOf=CN=ROLE_ADMIN,OU=OPENKM,DC=company,DC=com
        mail=user2@mail.com
        cn=User Name 3
    ou=organization2
      sAMAccountName=user3
        memberOf=CN=ROLE_USER,OU=OPENKM,DC=company,DC=com
        mail=user3@mail.com
        cn=User Name 3
      sAMAccountName=user4
        memberOf=CN=ROLE_USER,OU=OPENKM,DC=company,DC=com
        mail=user4@mail.com
        cn=User Name 4
```

Valid groups:

- cn=ROLE_ADMIN,ou=OPENKM,dc=company,dc=com
- cn=ROLE_USER,ou=OPENKM,dc=company,dc=com
- cn=ROLE_XXXX,ou=OPENKM,dc=company,dc=com
- cn=ROLE_YYYY,ou=OPENKM,dc=company,dc=com

Valid users:

- cn=user1,ou=organization1,dc=company,dc=com

- `cn=user2,ou=organization1,dc=company,dc=com`
- `cn=user3,ou=organization2,dc=company,dc=com`
- `cn=user4,ou=organization2,dc=company,dc=com`



Any distinguished name includes by default `dc=company,dc=com`

OpenKM.xml

- Parameter **follow** indicates several domains servers working together (balanced).
- Users defined in any Active Directory node will be able to log in, because it has defined `DC=company,DC=com` as the base filter.
- Any user authenticated in Active Directory can log in because it has not any filtering clause.
- Groups read by OpenKM can be defined in an Active Directory node, because it has defined `DC=company,DC=com` as the base filter.

```
<security:authentication-manager alias="authenticationManager">
  <security:authentication-provider ref="ldapAuthProvider" />
</security:authentication-manager>

<beans:bean id="contextSource" class="org.springframework.security.ldap.DefaultSpringSecurityContextSource">
  <beans:constructor-arg value="ldap://192.168.xxx.xxx:389"/>
  <beans:property name="userDn" value="CN=Administrator,OU=OPENKM,DC=company,DC=com"/>
  <beans:property name="password" value="****"/>
  <beans:property name="baseEnvironmentProperties">
    <beans:map>
      <beans:entry>
        <beans:key>
          <beans:value>java.naming.referral</beans:value>
        </beans:key>
        <beans:value>follow</beans:value>
      </beans:entry>
    </beans:map>
  </beans:property>
</beans:bean>

<beans:bean id="ldapAuthProvider" class="org.springframework.security.ldap.authentication.LdapAuthenticationProvider">
  <beans:constructor-arg>
    <beans:bean class="org.springframework.security.ldap.authentication.BindAuthenticationHandler">
      <beans:constructor-arg ref="contextSource"/>
      <beans:property name="userSearch" ref="userSearch"/>
    </beans:bean>
  </beans:constructor-arg>
  <beans:constructor-arg>
    <beans:bean class="org.springframework.security.ldap.userdetails.DefaultLdapAuthoritiesPopulator">
      <beans:constructor-arg ref="contextSource"/>
      <beans:constructor-arg value="DC=company,DC=com"/>
      <beans:property name="groupSearchFilter" value="member={0}" />
      <beans:property name="groupRoleAttribute" value="cn"/>
      <beans:property name="searchSubtree" value="true" />
      <beans:property name="convertToUpperCase" value="false" />
      <beans:property name="rolePrefix" value="" />
    </beans:bean>
  </beans:constructor-arg>
</beans:bean>
```

```
<beans:bean id="userSearch" class="org.springframework.security.ldap.search.FilterB
  <beans:constructor-arg index="0" value="DC=company,DC=com" />
  <beans:constructor-arg index="1" value="sAMAccountName={0}" />
  <beans:constructor-arg index="2" ref="contextSource" />
  <beans:property name="searchSubtree" value="true" />
</beans:bean>
```

Configuration parameters

Go to **Administration > Configuration parameters**:

Field / Property	Type	Description
principal.adapter	String	<code>com.openkm.principal.LdapPrincipalAdapter</code>
system.login.lowercase	String	<code>true</code>
principal.ldap.server	String	<code>ldap://192.168.xxx.xxx:389</code>
principal.ldap.security.principal	String	<code>CN=Administrator,OU=OPENKM,DC=company,DC=com</code>
principal.ldap.security.credentials	String	<code>password</code>
principal.ldap.referral	String	<code>follow</code>
principal.ldap.users.from.roles	Boolean	<code>false</code>
principal.ldap.user.attribute	String	<code>sAMAccountName</code>
principal.ldap.user.search.base	List	<code>DC=company,DC=com</code>
principal.ldap.user.search.filter	String	<code>(objectclass=user)</code>
principal.ldap.username.attribute	String	<code>cn</code>
principal.ldap.username.search.base	String	<code>DC=company,DC=com</code>
principal.ldap.username.search.filter	String	<code>(&(objectClass=person)(sAMAccountName={0}))</code>
principal.ldap.mail.attribute	String	<code>mail</code>

<i>principal.ldap.mail.search.base</i>	<i>String</i>	<i>DC=company,DC=com</i>
<i>principal.ldap.mail.search.filter</i>	<i>String</i>	<i>(&(objectClass=person)(sAMAccountName={0}))</i>
<i>principal.ldap.role.attribute</i>	<i>String</i>	<i>cn</i>
<i>principal.ldap.role.search.base</i>	<i>List</i>	<i>DC=company,DC=com</i>
<i>principal.ldap.role.search.filter</i>	<i>String</i>	<i>(objectclass=group)</i>
<i>principal.ldap.roles.by.user.attribute</i>	<i>String</i>	<i>memberOf</i>
<i>principal.ldap.roles.by.user.search.base</i>	<i>String</i>	<i>DC=company,DC=com</i>
<i>principal.ldap.roles.by.user.search.filter</i>	<i>String</i>	<i>(&(objectClass=person)(sAMAccountName={0}))</i>
<i>principal.ldap.users.by.role.attribute</i>	<i>String</i>	<i>member</i>
<i>principal.ldap.users.by.role.search.base</i>	<i>String</i>	<i>DC=company,DC=com</i>
<i>principal.ldap.users.by.role.search.filter</i>	<i>String</i>	<i>(&(objectClass=group)(CN={0}))</i>

Active Directory example with login based on filtering users by roles

In this example, we'll see an Active Directory connection that allows connecting any Active Directory authenticated user member of `ROLE_USER` or `ROLE_ADMIN`.

LDAP structure

```
dc=com
  dc=company
    ou=OPENKM
      cn=ROLE_ADMIN
        member=okmAdmin
        member=user1
      cn=ROLE_USER
        member=user3
      cn=ROLE_XXXX
        member=user2
      cn=ROLE_YYYY
        member=user4
      ...
    ou=organization1
      sAMAccountName=okmAdmin
        memberOf=CN=ROLE_ADMIN,OU=OPENKM,DC=company,DC=com
        userPrincipalName=okmAdmin@mail.com
        cn=OpenKM Administrator
      sAMAccountName=user1
        memberOf=CN=ROLE_ADMIN,OU=OPENKM,DC=company,DC=com
        userPrincipalName=user1@mail.com
        cn=User Name 1
      sAMAccountName=user2
        memberOf=CN=ROLE_XXXX,OU=OPENKM,DC=company,DC=com
        userPrincipalName=user2@mail.com
        cn=User Name 3
    ou=organization2
      sAMAccountName=user3
        memberOf=CN=ROLE_USER,OU=OPENKM,DC=company,DC=com
        userPrincipalName=user3@mail.com
        cn=User Name 3
      sAMAccountName=user4
        memberOf=CN=ROLE_YYYY,OU=OPENKM,DC=company,DC=com
        userPrincipalName=user4@mail.com
        cn=User Name 4
```

Valid groups:

- `cn=ROLE_ADMIN,ou=OPENKM,dc=company,dc=com`
- `cn=ROLE_USER,ou=OPENKM,dc=company,dc=com`
- `cn=ROLE_XXXX,ou=OPENKM,dc=company,dc=com`
- `cn=ROLE_YYYY,ou=OPENKM,dc=company,dc=com`

Valid users:

- `cn=user1,ou=organization1,dc=company,dc=com`

- `cn=user2,ou=organization1,dc=company,dc=com`

Invalid users:

- `cn=user3,ou=organization2,dc=company,dc=com`
- `cn=user4,ou=organization2,dc=company,dc=com`



Any distinguished name includes by default `dc=company,dc=com` except users who are not a member of **ROLE_ADMIN** or **ROLE_USER**

OpenKM.xml

- Users created in any Active Directory node members of **ROLE_ADMIN** or **ROLE_USER** will be able to log in because it has defined `DC=company,DC=com` as the base filter, and as **user account filter**.
- Groups read by OpenKM can be created in an Active Directory node, because it has defined `DC=company,DC=com` as the base filter.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns:beans="http://www.springframework.org/schema/beans"
  xmlns:security="http://www.springframework.org/schema/security"
  xmlns:task="http://www.springframework.org/schema/task"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/security
    http://www.springframework.org/schema/security/spring-security.xsd
    http://www.springframework.org/schema/task
    http://www.springframework.org/schema/task/spring-task.xsd">

  <security:authentication-manager alias="authenticationManager">
    <security:authentication-provider ref="ldapAuthProvider" />
  </security:authentication-manager>

  <beans:bean id="contextSource" class="org.springframework.security.ldap.DefaultSpringSecurityContextSource">
    <beans:constructor-arg value="ldap://192.168.xxx.xxx"/>
    <beans:property name="userDn" value="CN=connect,OU=OpenKM,DC=company,DC=com"/>
    <beans:property name="password" value="****"/>
    <beans:property name="baseEnvironmentProperties">
      <beans:map>
        <beans:entry>
          <beans:key>
            <beans:value>java.naming.referral</beans:value>
          </beans:key>
          <beans:value>follow</beans:value>
        </beans:entry>
      </beans:map>
    </beans:property>
  </beans:bean>

  <beans:bean id="ldapAuthProvider" class="org.springframework.security.ldap.authentication.LdapAuthenticationProvider">
    <beans:constructor-arg>
      <beans:bean class="org.springframework.security.ldap.authentication.BindAuthenticationHandler">
        <beans:constructor-arg ref="contextSource"/>
        <beans:property name="userSearch" ref="userSearch"/>
      </beans:bean>
    </beans:constructor-arg>
  </beans:bean>
</beans:beans>
```

```

<beans:constructor-arg>
  <beans:bean class="org.springframework.security.ldap.userdetails.DefaultLdapAutho
    <beans:constructor-arg ref="contextSource"/>
    <beans:constructor-arg value="DC=company,DC=com"/>
    <beans:property name="groupSearchFilter" value="member={0}"/>
    <beans:property name="groupRoleAttribute" value="cn"/>
    <beans:property name="searchSubtree" value="true" />
    <beans:property name="convertToUpperCase" value="false" />
    <beans:property name="rolePrefix" value="" />
  </beans:bean>
</beans:constructor-arg>
</beans:bean>

<beans:bean id="userSearch" class="org.springframework.security.ldap.search.FilterBas
  <beans:constructor-arg index="0" value="DC=company,DC=com" />
  <beans:constructor-arg index="1" value="(&(&sAMAccountName={0}) (&(&memberOf=CN=RO
  <beans:constructor-arg index="2" ref="contextSource" />
  <beans:property name="searchSubtree" value="true" />
</beans:bean>

</beans:beans>

```

Configuration parameters

- User members of `ROLE_ADMIN` or `ROLE_USER` can be created in any active directory node, because it has defined `DC=company,DC=com` as base filter, **`principal.ldap.user.search.base=DC=company,DC=com`** and user search filter as **`principal.ldap.user.search.filter=(&(objectclass=user)(memberOf=CN=ROLE_ADMIN,OU=OpenKM,DC=company,DC=com).`**
- Groups can be created in any active directory node, because is has `DC=company,DC=com` as base filter, **`principal.ldap.role.search.base=DC=company,DC=com.`**
- All Active Directory groups will be listed because it has not applied any filter restriction **`principal.ldap.role.search.filter=(objectclass=group).`**
- Mail attribute used in this case is **`userPrincipalName`** (the most common LDAP attribute to get mail is **`mail`**) **`principal.ldap.mail.attribute=userPrincipalName.`**

Go to **Administration > Configuration parameters**:

Field / Property	Type	Description
<code>principal.adapter</code>	String	<code>com.openkm.principal.LdapPrincipalAdapter</code>
<code>system.login.lowercase</code>	String	<code>true</code>
<code>principal.ldap.server</code>	String	<code>ldap://192.168.xxx.xxx:389</code>
<code>principal.ldap.security.principal</code>	String	<code>CN=Administrator,OU=OPENKM,DC=company,DC=com</code>
<code>principal.ldap.security.credentials</code>	String	<code>password</code>

<i>principal.ldap.referral</i>	String	<i>follow</i>
<i>principal.ldap.users.from.roles</i>	Boolean	<i>false</i>
<i>principal.ldap.user.attribute</i>	String	<i>sAMAccountName</i>
<i>principal.ldap.user.search.base</i>	List	<i>DC=company,DC=com</i>
<i>principal.ldap.user.search.filter</i>	String	<i>(&(objectclass=user)((memberOf=CN=ROLE_ADMIN,OU=OpenKM,DC=company,DC=com)(memberOf=CN=ROLE_USER,OU=OpenKM,DC=company,DC=com))</i>
<i>principal.ldap.username.attribute</i>	String	<i>cn</i>
<i>principal.ldap.username.search.base</i>	String	<i>DC=company,DC=com</i>
<i>principal.ldap.username.search.filter</i>	String	<i>(&(objectClass=person)(sAMAccountName={0}))</i>
<i>principal.ldap.mail.attribute</i>	String	<i>userPrincipalName</i>
<i>principal.ldap.mail.search.base</i>	String	<i>DC=company,DC=com</i>
<i>principal.ldap.mail.search.filter</i>	String	<i>(&(objectClass=person)(sAMAccountName={0}))</i>
<i>principal.ldap.role.attribute</i>	String	<i>cn</i>
<i>principal.ldap.role.search.base</i>	List	<i>DC=company,DC=com</i>
<i>principal.ldap.role.search.filter</i>	String	<i>(objectclass=group)</i>
<i>principal.ldap.roles.by.user.attribute</i>	String	<i>memberOf</i>
<i>principal.ldap.roles.by.user.search.base</i>	String	<i>DC=company,DC=com</i>
<i>principal.ldap.roles.by.user.search.filter</i>	String	<i>(&(objectClass=person)(sAMAccountName={0}))</i>

<i>principal.ldap.users.by.role.attribute</i>	<i>String</i>	<i>member</i>
<i>principal.ldap.users.by.role.search.base</i>	<i>String</i>	<i>OU=OpenKM,DC=company,DC=com</i>
<i>principal.ldap.users.by.role.search.filter</i>	<i>String</i>	<i>(&(objectClass=group)(cn={0}))</i>

Active Directory basic configuration

This is the suggested configuration to be used when roles and users are both defined in the same node, otherwise, refer to another example.

LDAP structure

```
dc=com
  dc=company
    cn=users
      cn=ROLE_ADMIN
        member=okmAdmin
        member=user1
      cn=ROLE_USER
        member=user3
      cn=ROLE_XXXX
        member=user2
      cn=ROLE_YYYY
        member=user4
      ...
      sAMAccountName=okmAdmin
        memberOf=cn=ROLE_ADMIN,cn=users,dc=company,dc=com
        userPrincipalName=okmAdmin@mail.com
        cn=OpenKM Administrator
      sAMAccountName=user1
        memberOf=cn=ROLE_ADMIN,cn=users,dc=company,dc=com
        userPrincipalName=user1@mail.com
        cn=User Name 1
      sAMAccountName=user2
        memberOf=cn=ROLE_USER,cn=users,dc=company,dc=com
        userPrincipalName=user2@mail.com
        cn=User Name 3
```

OpenKM.xml

```
<security:ldap-server id="ldapServer"
  url="ldap://192.168.0.6:389/DC=ldap,dc=company,dc=local"
  manager-dn="CN=Administrator,cn=users,dc=company,dc=local"
  manager-password="password"/>

<security:authentication-manager alias="authenticationManager">
  <security:ldap-authentication-provider
    server-ref="ldapServer"
    user-search-base="cn=Users"
    user-search-filter="(sAMAccountName={0}) "
    group-search-base="cn=Users"
    group-search-filter="(member={0}) "
    group-role-attribute="cn"
    role-prefix="none">
  </security:ldap-authentication-provider>
</security:authentication-manager>
```

Configuration parameters

- User members of ROLE_ADMIN or ROLE_USER are created into users node (these should be the distinguished names

CN=ROLE_ADMIN,cn=users,DC=company,DC=com and CN=ROLE_USER,cn=users,DC=company,DC=com).

- Groups can be created on any Active Directory node, because **DC=company,DC=com** is set as base filter,
principal.ldap.role.search.base=DC=company,DC=com.
- All Active Directory groups will be listed because it has not applied any filter restriction
principal.ldap.role.search.filter=(objectclass=group).

Go to Administration > Configuration parameters:

Field / Property	Type	Description
principal.adapter	String	<code>com.openkm.principal.LdapPrincipalAdapter</code>
system.login.lowercase	String	<code>true</code>
principal.ldap.server	String	<code>ldap://192.168.0.6:389</code>
principal.ldap.security.principal	String	<code>CN=Administrator,cn=users,dc=company,dc=local</code>
principal.ldap.security.credentials	String	<code>password</code>
principal.ldap.referral	String	
principal.ldap.users.from.roles	Boolean	<code>false</code>
principal.ldap.user.attribute	String	<code>sAMAccountName</code>
principal.ldap.user.search.base	List	<code>cn=users,dc=company,dc=local</code>
principal.ldap.user.search.filter	String	<code>(objectclass=person)</code>
principal.ldap.username.attribute	String	<code>cn</code>
principal.ldap.username.search.base	String	<code>cn=users,dc=company,dc=local</code>
principal.ldap.username.search.filter	String	<code>(&(objectClass=person)(sAMAccountName={0}))</code>
principal.ldap.mail.attribute	String	<code>mail</code>
principal.ldap.mail.search.base	String	<code>cn=users,dc=company,dc=local</code>

<i>principal.ldap.mail.search.filter</i>	<i>String</i>	<code>(&(objectClass=person)(sAMAccountName={0}))</code>
<i>principal.ldap.role.attribute</i>	<i>String</i>	<code>cn</code>
<i>principal.ldap.role.search.base</i>	<i>List</i>	<code>cn=users,dc=company,dc=local</code>
<i>principal.ldap.role.search.filter</i>	<i>String</i>	<code>(objectclass=group)</code>
<i>principal.ldap.roles.by.user.attribute</i>	<i>String</i>	<code>memberOf</code>
<i>principal.ldap.roles.by.user.search.base</i>	<i>String</i>	<code>DC=company,DC=com</code>
<i>principal.ldap.roles.by.user.search.filter</i>	<i>String</i>	<code>(&(objectClass=person)(sAMAccountName={0}))</code>
<i>principal.ldap.users.by.role.attribute</i>	<i>String</i>	<code>member</code>
<i>principal.ldap.users.by.role.search.base</i>	<i>String</i>	<code>cn={0},cn=users,dc=company,dc=local</code>
<i>principal.ldap.users.by.role.search.filter</i>	<i>String</i>	<code>(objectclass=group)</code>

Active Directory advanced configuration

LDAP structure

```
dc=com
  dc=company
    cn=users
      cn=ROLE_ADMIN
        member=okmAdmin
        member=user1
      cn=ROLE_USER
        member=user3
      cn=ROLE_XXXX
        member=user2
      cn=ROLE_YYYY
        member=user4
      ...
      sAMAccountName=okmAdmin
        memberOf=cn=ROLE_ADMIN,cn=users,dc=company,dc=com
        userPrincipalName=okmAdmin@mail.com
        cn=OpenKM Administrator
      sAMAccountName=user1
        memberOf=cn=ROLE_ADMIN,cn=users,dc=company,dc=com
        userPrincipalName=user1@mail.com
        cn=User Name 1
      sAMAccountName=user2
        memberOf=cn=ROLE_USER,cn=users,dc=company,dc=com
        userPrincipalName=user2@mail.com
        cn=User Name 3
```

OpenKM.xml

```
<security:authentication-manager alias="authenticationManager">
  <security:authentication-provider ref="ldapAuthProvider" />
</security:authentication-manager>

<beans:bean id="contextSource" class="org.springframework.security.ldap.DefaultSpringSecurityContextSource">
  <beans:constructor-arg value="ldap://192.168.0.6:389/dc=company,dc=local"/>
  <beans:property name="userDn" value="CN=Administrator,cn=users,dc=company,dc=local"/>
  <beans:property name="password" value="password"/>
</beans:bean>

<beans:bean id="ldapAuthProvider" class="org.springframework.security.ldap.authentication.LdapAuthenticationProvider">
  <beans:constructor-arg>
    <beans:bean class="org.springframework.security.ldap.authentication.BindAuthenticationHandler">
      <beans:constructor-arg ref="contextSource"/>
      <beans:property name="userSearch" ref="userSearch"/>
    </beans:bean>
  </beans:constructor-arg>
  <beans:constructor-arg>
    <beans:bean class="org.springframework.security.ldap.userdetails.DefaultLdapAuthoritiesPopulator">
      <beans:constructor-arg ref="contextSource"/>
      <beans:constructor-arg value=""/>
      <beans:property name="groupSearchFilter" value="memberOf={1}"/>
      <beans:property name="groupRoleAttribute" value="cn"/>
      <beans:property name="searchSubtree" value="true" />
      <beans:property name="convertToUpperCase" value="false" />
      <beans:property name="rolePrefix" value=""/>
    </beans:bean>
  </beans:constructor-arg>
```

```

</beans:bean>

<beans:bean id="userSearch" class="org.springframework.security.ldap.search.FilterBas
  <beans:constructor-arg index="0" value="" />
  <beans:constructor-arg index="1" value="sAMAccountName={0}" />
  <beans:constructor-arg index="2" ref="contextSource" />
  <beans:property name="searchSubtree" value="true" />
</beans:bean>

```

Configuration parameters

- User members of `ROLE_ADMIN` or `ROLE_USER` are created into users node (these should be the distinguished names `CN=ROLE_ADMIN,cn=users,DC=company,DC=com` and `CN=ROLE_USER,cn=users,DC=company,DC=com`).
- Groups can be created on any Active Directory node, because `DC=company,DC=com` is set as base filter, `principal.ldap.role.search.base=DC=company,DC=com`.
- All Active Directory groups will be listed because has not applied any filter restriction `principal.ldap.role.search.filter=(objectclass=group)`.

Go to **Administration > Configuration parameters**:

Field / Property	Type	Description
<code>principal.adapter</code>	String	<code>com.openkm.principal.LdapPrincipalAdapter</code>
<code>system.login.lowercase</code>	String	<code>true</code>
<code>principal.ldap.server</code>	String	<code>ldap://192.168.0.6:389</code>
<code>principal.ldap.security.principal</code>	String	<code>CN=Administrator,cn=users,dc=company,dc=local</code>
<code>principal.ldap.security.credentials</code>	String	<code>password</code>
<code>principal.ldap.referral</code>	String	
<code>principal.ldap.users.from.roles</code>	Boolean	<code>false</code>
<code>principal.ldap.user.attribute</code>	String	<code>sAMAccountName</code>
<code>principal.ldap.user.search.base</code>	List	<code>dc=company,dc=local</code>
<code>principal.ldap.user.search.filter</code>	String	<code>(objectclass=person)</code>
<code>principal.ldap.username.attribute</code>	String	<code>cn</code>

<i>principal.ldap.username.search.base</i>	<i>String</i>	<i>dc=company,dc=local</i>
<i>principal.ldap.username.search.filter</i>	<i>String</i>	<i>(&(objectClass=person)(sAMAccountName={0}))</i>
<i>principal.ldap.mail.attribute</i>	<i>String</i>	<i>mail</i>
<i>principal.ldap.mail.search.base</i>	<i>String</i>	<i>dc=company,dc=local</i>
<i>principal.ldap.mail.search.filter</i>	<i>String</i>	<i>(&(objectClass=person)(sAMAccountName={0}))</i>
<i>principal.ldap.role.attribute</i>	<i>String</i>	<i>cn</i>
<i>principal.ldap.role.search.base</i>	<i>List</i>	<i>dc=company,dc=local</i>
<i>principal.ldap.role.search.filter</i>	<i>String</i>	<i>(objectclass=group)</i>
<i>principal.ldap.roles.by.user.attribute</i>	<i>String</i>	<i>memberOf</i>
<i>principal.ldap.roles.by.user.search.base</i>	<i>String</i>	<i>DC=company,DC=com</i>
<i>principal.ldap.roles.by.user.search.filter</i>	<i>String</i>	<i>(&(objectClass=person)(sAMAccountName={0}))</i>
<i>principal.ldap.users.by.role.attribute</i>	<i>String</i>	<i>member</i>
<i>principal.ldap.users.by.role.search.base</i>	<i>String</i>	<i>dc=company,dc=local</i>
<i>principal.ldap.users.by.role.search.filter</i>	<i>String</i>	<i>(&(objectClass=group)(cn={0}))</i>

LDAP best practices for filtering users and roles

Usually you only want to retrieve a subset of users and roles present in your LDAP, to be shown in user interface lists or be able to login into.

Example based on single LDAP group

- Create a group named **OpenKM** (**CN=OpenKM,CN=users,DC=company,DC=com**). It will be used for filtering **users and roles** purpose.
- Assign roles and users are members of OpenKM group.
- Goal, only users and roles which are members of OpenKM group will be displayed in user interface lists.

For filtering user interface:

Field / Property	Type	Description
principal.ldap.user.search.filter	String	<code>(&(objectclass=person) (memberOf=CN=OpenKM,CN=users,dc=company,dc=com,</code>
principal.ldap.role.search.filter	String	<code>(&(objectclass=group) (memberOf=CN=OpenKM,CN=users,dc=company,dc=com,</code>
principal.ldap.users.by.role.search.filter	String	<code>(&(objectclass=group)(cn={0}) (memberOf=CN=OpenKM,CN=users,dc=company,dc=com,</code>
principal.ldap.roles.by.user.search.filter	String	<code>(&(objectClass=person)(sAMAccountName={0}) (memberOf=CN=OpenKM,CN=users,dc=company,dc=com,</code>

For login restriction:

Modify OpenKM.xml with:

```
<module-option name="baseFilter">(&(sAMAccountName={0})(memberOf=CN=OpenKM,CN=users,DC=company,DC=com,
```

Example based on ROLE_USER and ROLE_ADMIN group

- Create group name **OpenKM** (**CN=OpenKM,CN=users,DC=company,DC=com**). It will be used for filtering **roles** purpose.
- Assign roles are members of OpenKM group.
- Goal, only roles with OpenKM role will be displayed in user interface lists.

- Goal, only users what are members of `ROLE_USER` or `ROLE_ADMIN` group will be displayed in user interface lists.

For filtering user interface:

Field / Property	Type	Description
<i>principal.ldap.user.search.filter</i>	String	<code>(& (objectclass=person) ((memberOf=cn=ROLE_USER,cn=users,dc=company,dc=com) (memberOf=cn=ROLE_ADMIN,cn=users,dc=company,dc=com)</code>
<i>principal.ldap.role.search.filter</i>	String	<code>(& (objectclass=group) (memberOf=cn=OpenKM,dc=company,dc=com))</code>
<i>principal.ldap.users.by.role.search.filter</i>	String	<code>(& (objectClass=group) (cn={0}) (memberOf=cn=OpenKM,cn=users,DC=company,DC=com,</code>
<i>principal.ldap.roles.by.user.search.filter</i>	String	<code>(& (objectClass=person) (sAMAccountName={0}) (memberOf=cn=OpenKM,cn=users,DC=company,DC=com,</code>

For login restriction:

Modify `OpenKM.xml` with:

```
(& (sAMAccountName={0}) (objectClass=person) (| (memberOf=CN=ROLE_USER,CN=users,dc=we
```



There are some characters restricted in XML, one of these is `&`. Should be used `&` in place you usually should use `&`, otherwise you will get an error on starting the application.

ClearOS LDAP configuration

Need to install the Certificate Manager and Directory Server modules.

- Create an **okmAdmin** user in ClearOS.
- Create **ROLE_ADMIN**, **ROLE_USER** groups in ClearOS, populate.

LDAP Structure

```
dc=local
  dc=mydomain
    ou=Accounts
      ou=Groups
        cn=ROLE_ADMIN
          member=okmAdmin
          member=user1
          member=user2
          objectClass=posixGroup
        cn=ROLE_USER
          member=user3
          member=user4
          objectClass=posixGroup
        cn=ROLE_XXXX
          objectClass=posixGroup
        cn=ROLE_YYYY
          objectClass=posixGroup
        ...

      ou=Users
uid=okmAdmin
  memberOf=CN=ROLE_ADMIN,ou=Groups,ou=Accounts,dc=mydomain,dc=local
  mail=okmAdmin@mail.com
  cn=OpenKM Administrator
  objectClass=inetOrgPerson
uid=user1
  memberOf=CN=ROLE_ADMIN,ou=Groups,ou=Accounts,dc=mydomain,dc=local
  mail=user1@mail.com
  cn=User Name 1
  objectClass=inetOrgPerson
uid=user2
  memberOf=CN=ROLE_ADMIN,ou=Groups,ou=Accounts,dc=mydomain,dc=local
  mail=user2@mail.com
  cn=User Name 3
  objectClass=inetOrgPerson
```



Any distinguished name include by default **dc=mydomain,dc=local**

OpenKM.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns:beans="http://www.springframework.org/schema/beans"
  xmlns:security="http://www.springframework.org/schema/security"
  xmlns:task="http://www.springframework.org/schema/task"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
```

```

http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/security
http://www.springframework.org/schema/security/spring-security.xsd
http://www.springframework.org/schema/task
http://www.springframework.org/schema/task/spring-ta

<!-- Security configuration -->
<security:authentication-manager alias="authenticationManager">
  <security:authentication-provider ref="ldapAuthProvider" />
</security:authentication-manager>

<beans:bean id="contextSource" class="org.springframework.security.ldap.DefaultSpringSecurityContextSource">
  <beans:constructor-arg value="ldaps://ldap.mydomain.local:636/dc=mydomain,dc=local">
    <beans:property name="userDn" value="cn=Manager,ou=Internal,dc=mydomain,dc=local">
      <beans:property name="password" value="pass1234"/>
    </beans:property>
  </beans:property>
</beans:bean>

<beans:bean id="ldapAuthProvider" class="org.springframework.security.ldap.authentication.LdapAuthenticationProvider">
<beans:constructor-arg>
  <beans:bean class="org.springframework.security.ldap.authentication.BindAuthenticationHandler">
    <beans:constructor-arg ref="contextSource"/>
    <beans:property name="userSearch" ref="userSearch"></beans:property>
  </beans:bean>
</beans:constructor-arg>
<beans:constructor-arg>
  <beans:bean class="org.springframework.security.ldap.userdetails.DefaultLdapAuthoritiesResolver">
    <beans:constructor-arg ref="contextSource"/>
    <beans:constructor-arg value="ou=groups,ou=Accounts"/>
    <beans:property name="groupSearchFilter" value="member={0}"/>
    <beans:property name="groupRoleAttribute" value="cn"/>
    <beans:property name="searchSubtree" value="true" />
    <beans:property name="convertToUpperCase" value="true" />
    <beans:property name="rolePrefix" value="" />
  </beans:bean>
</beans:constructor-arg>
</beans:bean>

<beans:bean id="userSearch" class="org.springframework.security.ldap.search.FilterBasedLdapUserSearch">
  <beans:constructor-arg index="0" value="ou=Users,ou=Accounts" />
  <beans:constructor-arg index="1" value="uid={0}" />
  <beans:constructor-arg index="2" ref="contextSource" />
  <beans:property name="searchSubtree" value="true" />
</beans:bean>
</beans:beans>

```

Configuration parameters

Go to Administration > [Configuration parameters](#)

Field / Property	Type	Description
<i>principal.adapter</i>	String	<code>com.openkm.principal.LdapPrincipalAdapter</code>
<i>principal.database.filter.inactive.users</i>	Boolean	<code>true</code>
<i>principal.hide.connection.roles</i>	Boolean	<code>false</code>

<i>system.login.lowercase</i>	String	<code>true</code>
<i>principal.ldap.server</i>	String	<code>ldaps://ldap.mydomain.local:636</code>
<i>principal.ldap.security.principal</i>	String	<code>cn=Manager,ou=Internal,dc=mydomain,dc=local</code>
<i>principal.ldap.security.credentials</i>	String	<code>pass1234</code>
<i>principal.ldap.referral</i>	String	
<i>principal.ldap.users.from.roles</i>	Boolean	<code>false</code>
<i>principal.ldap.user.attribute</i>	String	<code>uid</code>
<i>principal.ldap.user.search.base</i>	List	<code>ou=Users,ou=Accounts,dc=mydomain,dc=local</code>
<i>principal.ldap.user.search.filter</i>	String	<code>(objectClass=inetOrgPerson)</code>
<i>principal.ldap.username.attribute</i>	String	<code>cn</code>
<i>principal.ldap.username.search.base</i>	String	<code>ou=Users,ou=Accounts,dc=mydomain,dc=local</code>
<i>principal.ldap.username.search.filter</i>	String	<code>(&(objectclass=inetOrgPerson)(uid={0}))</code>
<i>principal.ldap.mail.attribute</i>	String	<code>mail</code>
<i>principal.ldap.mail.search.base</i>	String	<code>dc=company,dc=local</code>
<i>principal.ldap.mail.search.filter</i>	String	<code>(&(objectclass=inetOrgPerson)(uid={0}))</code>
<i>principal.ldap.role.attribute</i>	String	<code>cn</code>
<i>principal.ldap.role.search.base</i>	List	<code>ou=Groups,ou=Accounts,dc=mydomain,dc=local</code>
<i>principal.ldap.role.search.filter</i>	String	<code>objectClass=posixGroup</code>

<i>principal.ldap.roles.by.user.attribute</i>	String	<code>memberOf</code>
<i>principal.ldap.roles.by.user.search.base</i>	String	<code>ou=Users,ou=Accounts,dc=mydomain,dc=local</code>
<i>principal.ldap.roles.by.user.search.filter</i>	String	<code>(&(objectClass=person)(uid={0}))</code>
<i>principal.ldap.users.by.role.attribute</i>	String	<code>member</code>
<i>principal.ldap.users.by.role.search.base</i>	String	<code>ou=Groups,ou=Accounts,dc=mydomain,dc=local</code>
<i>principal.ldap.users.by.role.search.filter</i>	String	<code>(&(objectClass=posixGroup)(cn={0}))</code>

Create a Certificate

ClearOS by default only allows LDAPS.

You should create a certificate on ClearOS with the leftmost RDN of the Subject being the common name (FQDN) of your ClearOS server. Easy way is to only specify the common name, and when signing use a policy that is permissive.

You can create an alias (CNAME) called ldap if you wish.

```
$ openssl req -out ldap.csr -key private/sys-0-key.pem -new
$ openssl ca -policy policy_anything -days 3650 -out /etc/pki/CA/certs/ldap.mydomain.local.crt -infiles ldap.csr
```

Add the CA cert to the nssdb:

```
$ certutil -A -d /etc/pki/nssdb/ -n "CA certificate" -t "CT,," -a -i /etc/pki/CA/ca-cert.pem
```

The LDAP server certificate needs to be saved in the nssdb with the name of Server-Cert.

```
$ openssl pkcs12 -export -inkey /etc/pki/CA/private/sys-0-key.pem -in /etc/pki/CA/certs/ldap.mydomain.local.crt -out
/root/Server-Cert.p12 -nodes -name 'Server-Cert'
$ pk12util -i /root/Server-Cert.p12 -d /etc/pki/nssdb/
```

Edit the file /etc/openldap/slapd.conf:

```
$ vim /etc/openldap/slapd.conf
```

```
TLSCACertificatePath    /etc/pki/nssdb
```

```
TLSCertificateFile      Server-Cert
TLSVerifyClient         never
```

Allow LDAP in ClearOS to be queried by OpenKM - edit the end of slapd.conf thusly:

```
access to *
  by self write
  by peername.ip=127.0.0.1 read
  ...
  by peername.ip=<OpenKM's IP address> read
```

Copy ClearOS' public CA certificate to the OpenKM server and add it to OpenKM's keystore:

```
$ keytool -import -trustcacerts -keystore /opt/openkm-6.2.3-community/java/jre/lib/security/cacerts -alias myca -file
/etc/pki/tls/certs/myca.pem
```

LDAP troubleshooting

javax.naming.PartialResultException error

If you see an exception like this "**javax.naming.PartialResultException**" probably you can not use [Active Directory basic configuration](#).



javax.naming.PartialResultException: Unprocessed Continuation Reference(s); remaining name 'cn=users,dc=company,dc=com'

Solution:

Enable ldap referral property.

Go to **Administration > Configuration parameters**:

Field / Property	Type	Description
principal.ldap.referral	String	<i>follow</i>

Read these articles:

- [Referrals in the JNDI](#)
- [JNDI Implementor Guidelines for LDAP Service Providers](#)

Can't connect to LDAP server

If your LDAP server is configured under SSL then you should use ldaps://

Solution:

Go to **Administration > Configuration parameters**:

Change server connection URL to ldaps.

Field / Property	Type	Description
principal.ldap.server	String	<i>ldaps://192.168.0.6:389</i>

Slow login or not able to login

It may be a problem with LDAP DNS names resolution.

Solution:

To prevent this issue - specially on Windows OS Family - it's a good practice to add all subdomains in application server's

hosts file (/etc/hosts for linux or C:/Windows/system32/driver/etc/host for windows).

Example based with active directory (LDAP) with distinguished base name **dc=company,dc=com** and server at **SBSSERVER** domain server name.

```
10.10.1.2 company.com
10.10.1.2 SBSSERVER SBSSERVER.company.com
10.10.1.2 Schema.Configuration.company.com
10.10.1.2 Configuration.company.com
10.10.1.2 DomainDnsZones.company.com
10.10.1.2 ForestDnsZones.company.com
```

More information at [OpenKM forum](#).

Force all users to be members of ROLE_USER

In a huge user AD repositories, can not be practice for the AD administrator to adding a new group to all the users, in this case we can force across xml login configuration that any logged user be member of ROLE_USER. Must be added the xml tag

<beans:property name="defaultRole" value="ROLE_USER" />.

For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns:beans="http://www.springframework.org/schema/beans"
  xmlns:security="http://www.springframework.org/schema/security"
  xmlns:task="http://www.springframework.org/schema/task"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/security
    http://www.springframework.org/schema/security/spring-security.xsd
    http://www.springframework.org/schema/task/spring-task.xsd">

  <security:authentication-manager alias="authenticationManager">
    <security:authentication-provider ref="ldapAuthProvider" />
  </security:authentication-manager>

  <beans:bean id="contextSource" class="org.springframework.security.ldap.DefaultSpringSecurityContextSource">
    <beans:constructor-arg value="ldap://192.168.xxx.xxx"/>
    <beans:property name="userDn" value="CN=connect,OU=OpenKM,DC=company,DC=com"/>
    <beans:property name="password" value="****"/>
    <beans:property name="baseEnvironmentProperties">
      <beans:map>
        <beans:entry>
          <beans:key>
            <beans:value>java.naming.referral</beans:value>
          </beans:key>
          <beans:value>follow</beans:value>
        </beans:entry>
      </beans:map>
    </beans:property>
  </beans:bean>

  <beans:bean id="ldapAuthProvider" class="org.springframework.security.ldap.authentication.LdapAuthenticationProvider">
    <beans:constructor-arg>
```

```

    <beans:bean class="org.springframework.security.ldap.authentication.BindAuthentic
    <beans:constructor-arg ref="contextSource"/>
    <beans:property name="userSearch" ref="userSearch"/>
  </beans:bean>
</beans:constructor-arg>
<beans:constructor-arg>
  <beans:bean class="org.springframework.security.ldap.userdetails.DefaultLdapAutho
  <beans:constructor-arg ref="contextSource"/>
  <beans:constructor-arg value="DC=company,DC=com"/>
  <beans:property name="groupSearchFilter" value="member={0}"/>
  <beans:property name="groupRoleAttribute" value="cn"/>
  <beans:property name="searchSubtree" value="true" />
  <beans:property name="convertToUpperCase" value="false" />
  <beans:property name="rolePrefix" value="" />
  <beans:property name="defaultRole" value="ROLE_USER" />
  </beans:bean>
</beans:constructor-arg>
</beans:bean>

<beans:bean id="userSearch" class="org.springframework.security.ldap.search.FilterBas
  <beans:constructor-arg index="0" value="DC=company,DC=com" />
  <beans:constructor-arg index="1" value="( & (sAMAccountName={0}) (| (memberOf=CN=RO
  <beans:constructor-arg index="2" ref="contextSource" />
  <beans:property name="searchSubtree" value="true" />
</beans:bean>

</beans:beans>

```

org.springframework.beans.factory.parsing.BeanDefinitionParsingException

If you see an error like this "**org.springframework.beans.factory.parsing.BeanDefinitionParsingException**" probably you have some spring wrong version set in the OpenKM.xml:



org.springframework.beans.factory.parsing.BeanDefinitionParsingException: Configuration problem: You cannot use a spring-security-2.0.xsd or spring-security-3.0.xsd or spring-security-3.1.xsd schema with Spring Security 3.2. Please update your schema declarations to the 3.2 schema.
 Offending resource: URL [file:/desarrollo/tomcat/tomcat-7.0.61-openkm-6.4/OpenKM.xml]
 at
 org.springframework.beans.factory.parsing.FailFastProblemReporter.fatal(FailFastProblemReporter.java:59)

Solution:

Edit the OpenKM.xml file and remove the version numbers:

```

<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns:beans="http://www.springframework.org/schema/beans"
  xmlns:security="http://www.springframework.org/schema/security"
  xmlns:task="http://www.springframework.org/schema/task"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-b
    http://www.springframework.org/schema/security
    http://www.springframework.org/schema/security/sprin
    http://www.springframework.org/schema/task
    http://www.springframework.org/schema/task/spring-ta

```


to

```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns:beans="http://www.springframework.org/schema/beans"
  xmlns:security="http://www.springframework.org/schema/security"
  xmlns:task="http://www.springframework.org/schema/task"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/security
    http://www.springframework.org/schema/security/spring-security.xsd
    http://www.springframework.org/schema/task
    http://www.springframework.org/schema/task/spring-ta
```

LDAP: error code 8 - BindSimple: Transport encryption required

The transport encryption required error usually happens when you are trying to connect with ldaps (with SSL encryption) and you have not configured the trust certificate into the JAVA certs storage.

First, check you are using "ldaps://" - ends with s - in your **OpenKM.xml** file for the connection, when you get this error you should use it.

The first step is to **get the certificate from the server** for it follow the steps below:

```
$ openssl s_client -showcerts -connect your.ldap.server.com:636
```

The output will contain a number of entries delimited with:

```
-----BEGIN CERTIFICATE-----
aklfhskfadljasdl1340234234ASDSDFSDFSDFSDFS
....
-----END CERTIFICATE-----
```

Copy the last certificate entry into a file (ldapca.crt)

Then, **add it to the java** keystore in \$JRE_HOME/lib/security

```
$ cd $JRE_HOME/lib/security
$ keytool -import -alias ldapca_self_sign -keystore cacerts -storepass changeit -file ldapca.crt
```

Finally **enable trust store** into the file \$TOMCAT_HOME/bin/setenv.sh

```
# JAVA enable trust store
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStore=/etc/pki/java/cacerts -Djavax.net.s
```



Additional information:

- <https://docs.spring.io/spring-ldap/docs/1.2.0/reference/#dir-context-url>

- <https://stackoverflow.com/questions/9903560/spring-security-ssl-ldap-and-no-certificate>

Ldap error: Caused by java.security.cert.CertificateException: No subject alternative DNS name matching

In the log maybe show some stack trace error like:

```
Caused by: java.security.cert.CertificateException: No subject alternative DNS name m
    at sun.security.util.HostnameChecker.matchDNS (HostnameChecker.java:214)
    at sun.security.util.HostnameChecker.match (HostnameChecker.java:96)
    at sun.security.ssl.X509TrustManagerImpl.checkIdentity (X509TrustManagerImpl.j
    at sun.security.ssl.X509TrustManagerImpl.checkIdentity (X509TrustManagerImpl.j
    at sun.security.ssl.X509TrustManagerImpl.checkTrusted (X509TrustManagerImpl.ja
    at sun.security.ssl.X509TrustManagerImpl.checkServerTrusted (X509TrustManagerI
    at sun.security.ssl.ClientHandshaker.serverCertificate (ClientHandshaker.java:
    ... 72 more
```

Or like:

```
Caused by: org.springframework.ldap.CommunicationException: simple bind failed: ldap
    at org.springframework.ldap.support.LdapUtils.convertLdapException (LdapUtils.
    at org.springframework.ldap.core.support.AbstractContextSource.createContext (.
    at org.springframework.ldap.core.support.AbstractContextSource.doGetContext (A
    at org.springframework.ldap.core.support.AbstractContextSource.getReadOnlyCon
    at org.springframework.ldap.core.LdapTemplate.executeReadOnly (LdapTemplate.ja
    at org.springframework.security.ldap.SpringSecurityLdapTemplate.searchForSing
    at org.springframework.security.ldap.search.FilterBasedLdapUserSearch.searchF
    at org.springframework.security.ldap.authentication.BindAuthenticator.authenti
    at org.springframework.security.ldap.authentication.LdapAuthenticationProvide
```

Modify JVM configuration parameters in `setenv.sh` or `setenv.bat` adding the JVM configuration parameter:

```
-Dcom.sun.jndi.ldap.object.disableEndpointIdentification=true
```

Log Configuration

Log4j based Log Configuration

OpenKM uses log4j which is a Java-based logging utility. This is useful, for example when you need to get more information about something that is not working properly in OpenKM, for example third-party application configuration or Spring authentication. By default only log messages with INFO level are shown, but you can increase this log information by raising the log level to WARN or DEBUG. The log configuration file is **\$TOMCAT_HOME/conf/log4j.properties** here you can change the error level message generated by a give class or package.



When OpenKM detect a modification in the **log4j.properties** file, it will reload it so you don't need to stop and start OpenKM to make these changes effective.

Third-party application execution

This is useful when something like a document preview fails, because this way OpenKM will log every external program execution:

```
log4j.logger.com.openkm.util.ExecutionUtils=DEBUG
```

Enable log authentication

If you need to debug the security configuration, edit the **log4j.properties** file and add this line:

```
log4j.logger.org.springframework.security=DEBUG
```

OpenOffice and LibreOffice configuration



After changing **system.openoffice.path** parameter is required to restart application.

Linux

Debian & Ubuntu



OpenKM has been roughly tested with recent build of **OpenOffice** and **LibreOffice**, but in recent Linux distros only **LibreOffice** is available to install. Anyway it works pretty well.

Install application:

```
$ apt-get install libreoffice
```

Install languages:

```
$ apt-get install libreoffice-l10n-en-gb
```

Go to **Administration > Configuration parameters**:

Field / Property	Type	Description
system.openoffice.path	String	<div> The path may change depending your OS version. To verify that your path is correct, take a look into your chosen path and it should be a folder named program, otherwise the path is not correct.</div> <div><code>/usr/bin/libreoffice</code></div>

CentOS

Install the application:

```
$ yum install libreoffice
```

Install languages:

```
$ yum install libreoffice-langpack-en
```



By default, OpenOffice or LibreOffice is not present on Red Hat repository. If you're using this OS then you should try to install manually.

Download and install one of them from:

- LibreOffice: <http://www.libreoffice.org>
- OpenOffice: <http://www.openoffice.org>

Check if you've installed all needed packages:

```
$ rpm -qa | grep openoffice
```

These packages are mandatory:

- **openoffice.org-ure-3.1.1-19.5.el5_5.6** (ure)
- **openoffice.org-headless-3.1.1-19.5.el5_5.6** (headless)
- **openoffice.org-pyuno-3.1.1-19.5.el5_5.6** (pyuno)



The name of the packages can change depending your OpenOffice version. Take these as reference for your check.

Go to **Administration > Configuration parameters**:

Field / Property	Type	Description
system.openoffice.path	String	<div> <p>The path can change depending your OS version. To verify your path is correct, into your chosen path should be a folder named program, otherwise the path is not correct.</p> </div> <div> <pre>/usr/bin/libreoffice</pre> </div>

Windows

Download and install one of them:


- LibreOffice: <http://www.libreoffice.org>
- OpenOffice: <http://www.openoffice.org>





Our suggestion is to use OpenOffice in Windows OS. In some Windows OS we've found unexpected problems that are automatically solved changing LibreOffice to OpenOffice. Take it only as a suggestion.

Go to **Administration > Configuration parameters**:

Field / Property	Type	Description
		<div> <p>The path can change depending your OS version. To verify your path</p> </div>

system.openoffice.path	String	 is correct, into your chosen path should be a folder named program, otherwise the path is not correct.
		c:\Program files\OpenOffice 4

Check configuration

- Go to Administration > Configuration parameters and click on check icon . If all is right you should see under **system.openoffice.path** parameter an **OK** status.
- After application starts you should see a process called **soffice.bin**.
- Upload a MS Office or OpenOffice file. Select the document and from toolbar choose the option **Download as PDF** .
- Take a look at \$TOMCAT_HOME/logs/catalina.log file. At the end of the startup process you should see something like this:

```

2015-03-17 17:10:45,792 [pool-2-thread-1] INFO com.openkm.servlet.RepositoryStartupS
2015-03-17 17:10:45,813 [pool-2-thread-1] INFO com.openkm.util.DocConverter - *** Bu
2015-03-17 17:10:45,813 [pool-2-thread-1] INFO com.openkm.util.DocConverter - system
2015-03-17 17:10:45,813 [pool-2-thread-1] INFO com.openkm.util.DocConverter - system
2015-03-17 17:10:45,813 [pool-2-thread-1] INFO com.openkm.util.DocConverter - system
2015-03-17 17:10:45,866 [pool-2-thread-1] INFO org.artofsolving.jodconverter.office..
2015-03-17 17:10:45,915 [OfficeProcessThread-0] WARN org.artofsolving.jodconverter.o
2015-03-17 17:10:45,934 [OfficeProcessThread-0] INFO org.artofsolving.jodconverter.o
2015-03-17 17:10:45,956 [OfficeProcessThread-0] INFO org.artofsolving.jodconverter.o
2015-03-17 17:10:48,818 [OfficeProcessThread-0] WARN org.artofsolving.jodconverter.o
2015-03-17 17:10:48,825 [OfficeProcessThread-0] INFO org.artofsolving.jodconverter.o
2015-03-17 17:10:48,832 [OfficeProcessThread-0] INFO org.artofsolving.jodconverter.o

```



Read [OpenOffice LibreOffice configuration parameters](#) to get more information about all available OpenOffice & LibreOffice configuration parameters.

Security configuration

OpenKM can work with several security managers, depending on the selected value, the manner in which the result is evaluated is different. This application has separated managers, one for the search engine and other for the rest of the application.

Search engine managers:

- *lucene*.
- *am_more*, *am_window* or *am_limited*.



The fastest one is **lucene**, because it is the Lucene search engine which restricts the search results based on the node security. The others are **am_more**, **am_window** and **am_limited**, which make use of the AccessManager - `security.access.manager` value - to restrict the search results, these will reduce OpenKM performance (at least in search) but will match customer requirements. The performance can be improved customizing `com.openkm.cache.parentNodePermissions` cache.

Rest of application manager:

- *simple*.
- *recursive*.
- *read_recursive*.







The **simple** option only evaluates the selected node. With **recursive**, the security is evaluated starting with the selected node until the repository root (so, evaluating from bottom to top). The **read_recursive** is used to prevent showing documents in search results when any folder in the path doesn't have read access for the user who ran the query. Should be used in combination with **am_more**.

Application by default is configured with:

Field / Property	Type	Description
<code>security.access.manager</code>	<i>String</i>	<code>simple</code>
<code>security.search.evaluation</code>	<i>String</i>	<code>lucene</code>

Differences between simple and recursive

- `okm:root`
 - `level-1a`
 - `level-2a`
 - `level-3a`

-  alpha.doc
 -  beta.doc
 -  gamma.doc
-  level-3b
-  level-1b
-  level-1c

When using simple if you revoke permission from the level-2a folder and the user tries to view the level-3a folder, OpenKM will still show the documents to this user. When the user performs a search, OpenKM will return these documents as search results. However, the user can't access the documents because when OpenKM tries to access the document, it will open every path location and will fail to open level-2a because the user has no rights. But these document can be accessed using the API. You need to apply revoke recursively to prevent the user from seeing these documents, so they won't be shown in the search results.

When using recursive if you revoke permission from the level-2a folder and the user tries to view the level-3a folder, OpenKM will report an error because Access Manager also evaluates the permission of the level-2a ancestor node. When the user performs a search, OpenKM won't show these documents in the search results. Keep in mind that this Access Manager implementation is slower because it needs to evaluate more nodes: the more depth node the more parent nodes need to be evaluated.

Extending security

You can use the extend security to download files, start workflows, add, remove or modify property groups (metadata) or compact history. To enable extended security evaluation you must change the value of **security.extended.mask** parameter.



After the value is changed, refresh the browser to take effect in user view.

Values:

- `DOWNLOAD` = 1024;
- `START_WORKFLOW` = 2048;
- `COMPACT_HISTORY` = 4096;
- `PROPERTY_GROUP` = 8192;

For example, to enable download and property groups you should put the mask 9216 (1024 + 8192)

Field / Property	Type	Description
<code>security.extended.mask</code>	Integer	9216

Configuration for security changes which take a long time

By default when you change the security of a node in standard view, every time you click to grant or revoke a permission the action is performed by OpenKM. This means that if you want to change the security of 10 users, 10 commands are sent to the server. This became a problem when every change affects several hundred (or even thousands) of nodes. There's an alternative security configuration parameter named **security.mode.multiple** which enables multiple changes at a time, so it's so faster.

Field / Property	Type	Description
security.mode.multiple	Boolean	<div>true</div>

Cache configuration

OpenKM uses several caches to enhance performance. Some of these caches are directly used by OpenKM and others indirectly, for example those used by Hibernate. All these caches have default values which make them useful but these default values are not always good for every installation. The cache engine used is the very popular [EHCACHE](#), which is widely used and very configurable. You can see the default values at `$TOMCAT_HOME/webapps/OpenKM/WEB-INF/classes/ehcache.xml`.



Don't modify this file because all changes will be lost when OpenKM is restarted or updated.

If you want to tweak this default configuration you have to duplicate the provided configuration file into `$TOMCAT_HOME/ehcache.xml` and make the proper changes to match your requirements. This is a sample definition:

```
<cache name="com.openkm.cache.ldapPrincipalAdapter.general"
      maxElementsInMemory="1000"
      timeToIdleSeconds="300"
      timeToLiveSeconds="0"
      overflowToDisk="true"
      eternal="false" />
```

The most important values are:

- **timeToLive:** The maximum number of seconds an element can exist in the cache regardless of use. The element expires at this limit and will no longer be returned from the cache. The default value is 0, which means no TTL eviction takes place (infinite lifetime).
- **timeToIdle:** The maximum number of seconds an element can exist in the cache without being accessed. The element expires at this limit and will no longer be returned from the cache. The default value is 0, which means no TTI eviction takes place (infinite lifetime).



Be careful with **eternal**, because it sets whether elements are eternal. In case of true, timeouts are ignored and the element never expires.

So, in this case the cache called "com.openkm.cache.ldapPrincipalAdapter.general" will have a maximum of 1000 elements in memory and in case of overflow will be written to disk. These elements will be removed from this cache if are unused for 5 minutes. Otherwise, will remain forever. If you want to force a refresh, you have to change the **timeToLiveSeconds** to 3600 and will be refreshed every hour.

You can check these caches status by going to **Administration > Utilities > Cache stats**. In this part you can enable and see the cache usage stats and even reset a give one. More info at [Cache stats](#).

Throubleshooting

Here you can find some common problems and how to solve them. The context is grouped into several sections:

- [Server configuration](#)
- [Client configuration](#)
- [Browser configuration](#)

*In special, take a look to **Java and Chrome problem** section into [Browser configuration](#) which will help you to get Java applets working in Google Chrome.*

Server configuration

Packet for query is too large (MySQL)

If you get an error like this:

Packet for query is too large (1708726 > 1048576).
You can change this value on the server by setting the 'max_allowed_packet' variable.

You need to modify your MySQL server configuration file and increase the value of the **max_allowed_packet** property. Don't forget to restart the MySQL after any change in this file. In Debian based distros like Ubuntu, this configuration file is located at `/etc/mysql/my.cnf`.

Not able to connect from other computer in the network



Ensure you have access from the other computers to the server.
Ensure your server firewall allows remote connection to the destination port.

If you expose the tomcat port 8080, ensure is binding your ip, the **"address"** parameter value should be **"0.0.0.0"** at `$TOMCAT_HOME/conf/server.xml`:

```
<Connector port="8080" address="0.0.0.0" protocol="HTTP/1.1"
  connectionTimeout="20000"
  redirectPort="8443" />
```



After any modification in `server.xml` to take effect must restart the tomcat.

Error downloading document as PDF

This error usually is caused by OpenOffice or LibreOffice service configuration.



Check if the OpenOffice or LibreOffice service is started by OpenKM. You should get a program named **"soffice.bin"** running after application finished the startup.

IMAP SSL certificate error

First obtain the certificate with the `openssl` command and after this import the certificate with:

```
$ keytool -import -alias mail.yourcompany.com -keystore /etc/java-6-sun/security/cacerts -file imapd.pem
```



The application must be restarted to refresh the SSL certificates. More information at [Connecting+to+SSL+services](#).

OpenKM server has no direct Internet connection

If you need a proxy to access the Internet, edit your `"/etc/bash.bashrc"` file as root and put these lines at the end of your `"/etc/bash.bashrc"` file:

```
export http_proxy=http://username:password@proxyserver.net:port/  
export ftp_proxy=http://username:password@proxyserver.net:port/
```

You can omit the `username:password`, if your proxy server has no password.

Access denied installing OpenKM as service on Windows 7

The command shell works slightly differently in Vista. By default when you run `cmd.exe`, it does not give administrator rights. To get around it, **create a shortcut to `cmd.exe` on desktop, open Properties > Compatibility tab and check box for Administrator rights**. Then you can do admin stuff on `cmd` line.

Problems with character encodings

If your language has non-ASCII characters and they are displayed as garbage you have a problem with encodings.

Check the `$TOMCAT/bin/setenv.sh` or `$TOMCAT/bin/setenv.bat` file. You should have set the JVM parameter `"-Dfile.encoding=utf-8"`.



The application must be restarted.

PermGen errors



Since Java 8 there is no such **PermSize** and **MaxPermSize** options.

If you see this error:

```
java.lang.OutOfMemoryError: PermGen space
```

Edit your `"$TOMCAT_HOME/bin/setenv.sh"` (or `"$TOMCAT_HOME/bin/setenv.bat"` if using Windows) and increase the value of the JVM parameter `"-XX:MaxPermSize"`:

```
JAVA_OPTS="-Xms256m -Xmx2048m -XX:PermSize=128m -XX:MaxPermSize=512m"
```

See also [Presenting the Permanent Generation](#).

LDAP Sizerlimit exceeded

Caused because bindDN has size limit for each ldap query.

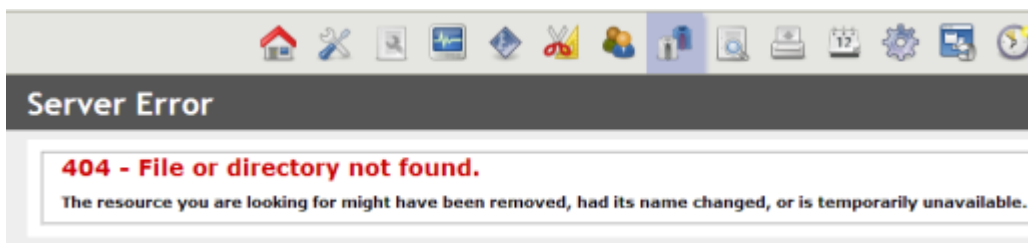
In Openldap The solution is to increase the limit of bindDN, for example (to change size limit to user used by openkm ldap connection)

```
limits dn.exact="cn=reader,ou=accounts,ou=admins,dc=example,dc=com" size=100000
```

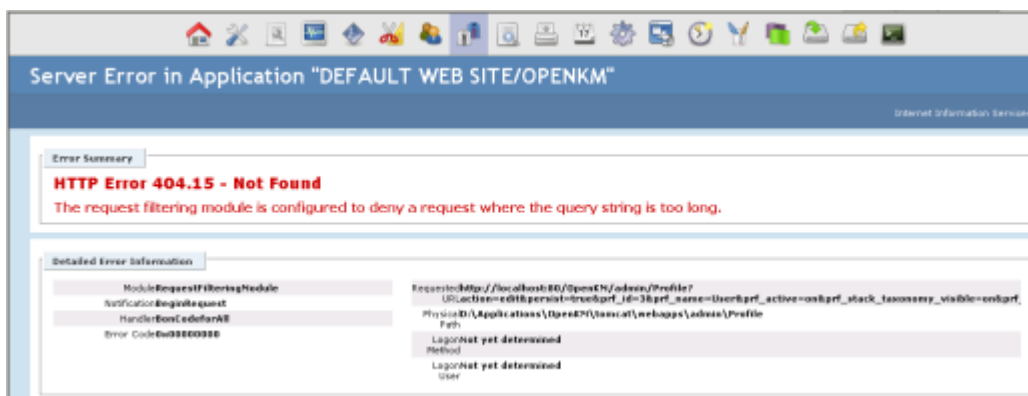
See also [OpenLDAP Limits](#).

Behind IIS proxy you get error 404 or 404.15

Using IIS as a web server by default it has a limit for the query string sent to the server if limit is passed then you get some error screen on the client side:

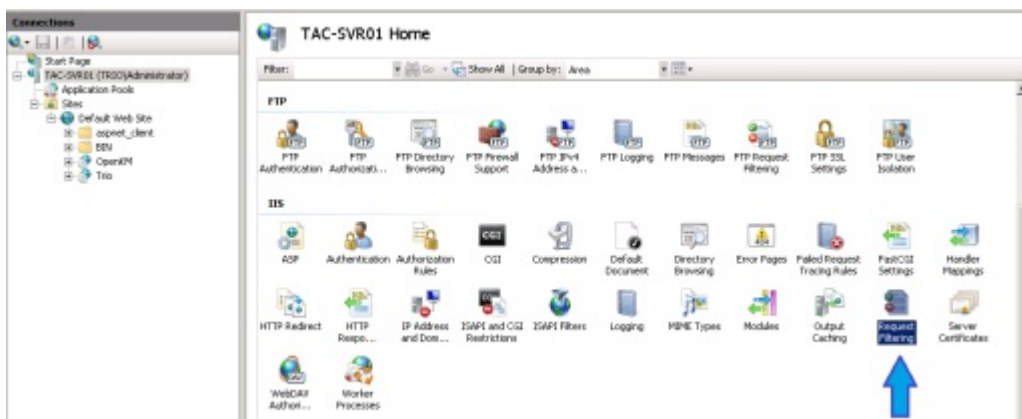


In the server side we get a more detailed error:



To solve this issue we need to increase the maximum size of the query string on the web server:

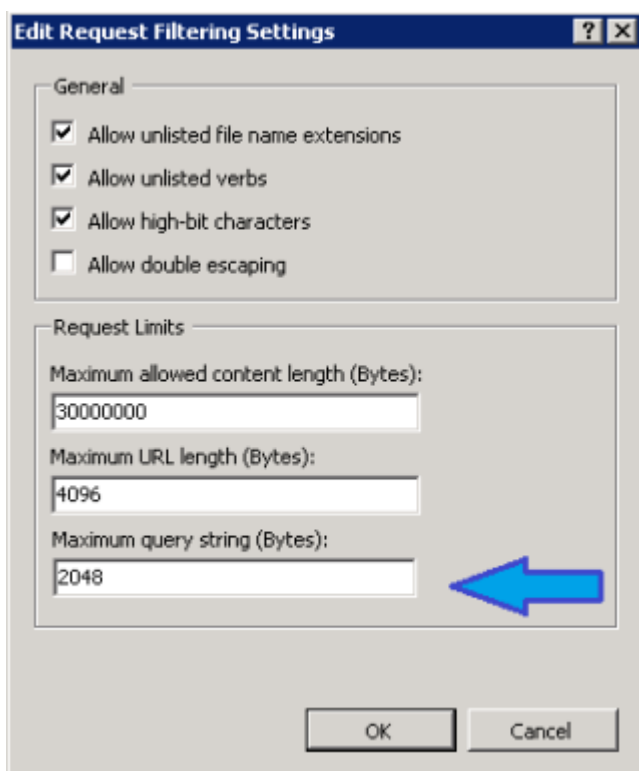
- **Open Internet Information Services (IIS) Manager.**
- In the **Connections pane**, go to the **connection > site > application**, or directory for which you want to modify your request filtering settings.
- In the **Home pane**, double-click **Request Filtering**.



Click **Edit Feature Settings...** in the Actions pane.



Change the maximum query string length:

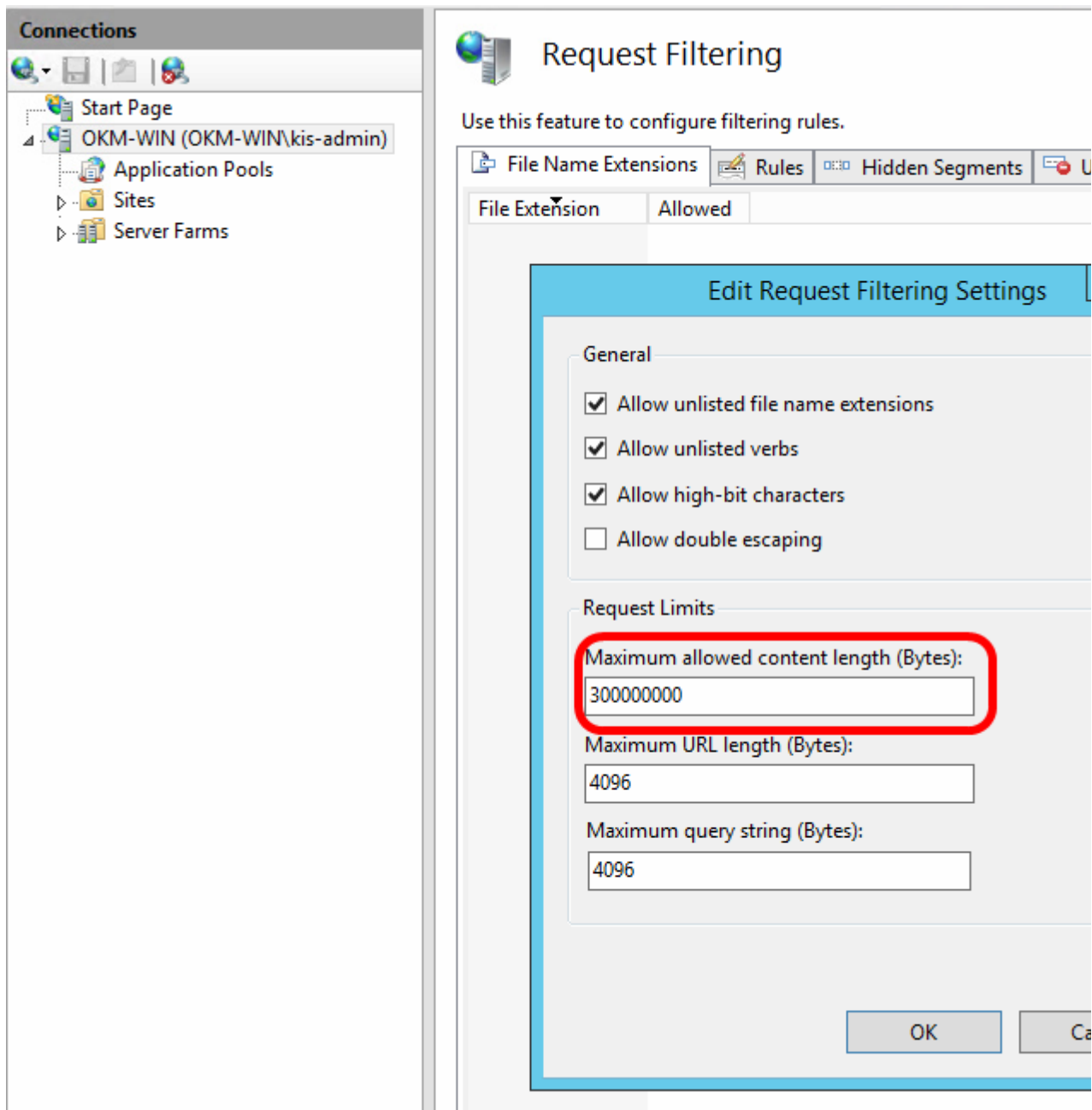


More information at [OpenKM forum](#).

IIS gets IOException: Internal Server Error while uploading file

Solution one

Change the "Maximum allowed content length (Bytes)" from your IIS configuration parameters:



Solution two

- Go to webapps folder
- Edit the web.config file and add the following under <configuration> where "2097151" is the maximum allowed in IIS:


```
<system.web>
  <httpRuntime maxRequestLength="2097151" />
</system.web>
```

If there is no web.config then just create one from text file and add the following :

```
<configuration>
  <system.web>
    <httpRuntime maxRequestLength="2097151" />
  </system.web>
</configuration>
```

Zoho fails with SSL connection

To use Zoho integration you have to configure your server to deliver the whole certificate chain by itself, because the Zoho client does not look for the chain. Our solution is reconfigure APR in Tomcat to deliver the CA chain separately to the server certificate.

The configuration file at "\$TOMCAT_HOME/conf/server.xml" it looks like this:

```
<!-- Define a SSL HTTP/1.1 Connector on port 8443
This connector uses the JSSE configuration, when using APR, the
connector should be using the OpenSSL style configuration
described in the APR documentation -->

<Connector port="8443" address="0.0.0.0" protocol="HTTP/1.1" SSLEnabled="true"
    maxThreads="150" scheme="https" secure="true" acceptCount="100"
    enableLookups="false" clientAuth="true" sslProtocol="SSLv3"
    SSLCipherSuite="ALL:!ADH:RC4+RSA:+HIGH:+MEDIUM:!LOW:!SSLv2:!EXPORT"
    SSLCertificateFile="/home/openkm/tomcat-7.0.27/cert/serverpublic.pem"
    SSLCertificateKeyFile="/home/openkm/tomcat-7.0.27/cert/serverprivate.pem"
    SSLCertificateChainFile="/home/openkm/tomcat-7.0.27/cert/serverpublicchain.pem"
    allowUnsafeLegacyRenegotiation="false"
/>
```



More information at our [Forum](#).

Sigar java.lang.UnsatisfiedLinkError Known issue in Windows

If you have OpenOffice or LibreOffice configured and you get the error "**sigar java.lang.UnsatisfiedLinkError**" probably you have missing the "**sigar path**" in "**PATH**" environment variable.

```
Caused by: java.lang.UnsatisfiedLinkError: org.hyperic.sigar.Sigar.getSigNum(Ljava/lang/String;)I
    at org.hyperic.sigar.Sigar.getSigNum(Native Method)
    at org.hyperic.sigar.Sigar.kill(Sigar.java:275)
```

Edit your general environment variable "**PATH**" and add your "**\$TOMCAT_HOME\lib\sigar**" folder, for example like

PATH=...;C:\tomcat-7.0.27\lib\sigar



The application must be restarted.

WARN net.sf.ehcache.hibernate.strategy.AbstractReadWriteEhcacheAccessStrategy

If you see this message:

*A soft-locked cache entry was expired by the underlying Ehcache.
If this happens regularly you should consider increasing the cache timeouts and/or ca*

Do you experiencing any performance or usability problems as a result of this warning? If not then it seems like you should treat this warning as just that: a warning.

*Otherwise, it looks like the solution is to increase the size of **maxElementsInMemory** to account for the difference in what is cached versus what is realistically being used.*

OpenOffice is not starting on Windows

Sometimes is not easy to get OpenKM with soffice service started for several and strange reason which depends on your OS Windows version and other specific configuration done in server.

*First of all ensure your OpenOffice path is correctly configured at **Administration > Configuration parameters** > "system.openoffice.path" parameter.*

Ensure there's no other soffice.bin instance running. If there're another soffice.bin instance on execution openkm will not be able to start their own soffice service. Take a look at the process running on Widnows or use a tool like [Process Explorer](#).

Stop firewall and other possible tools which be able to do not allow start soffice service. For example the antivirus.

Try starting application from command line at "%TOMCAT_HOME/bin":

```
$ catalina.bat run
```



Another option is open a command line with administrator grants. In some Windows OS appears a checkbox to execute with some restrictions, we suggest try both options.

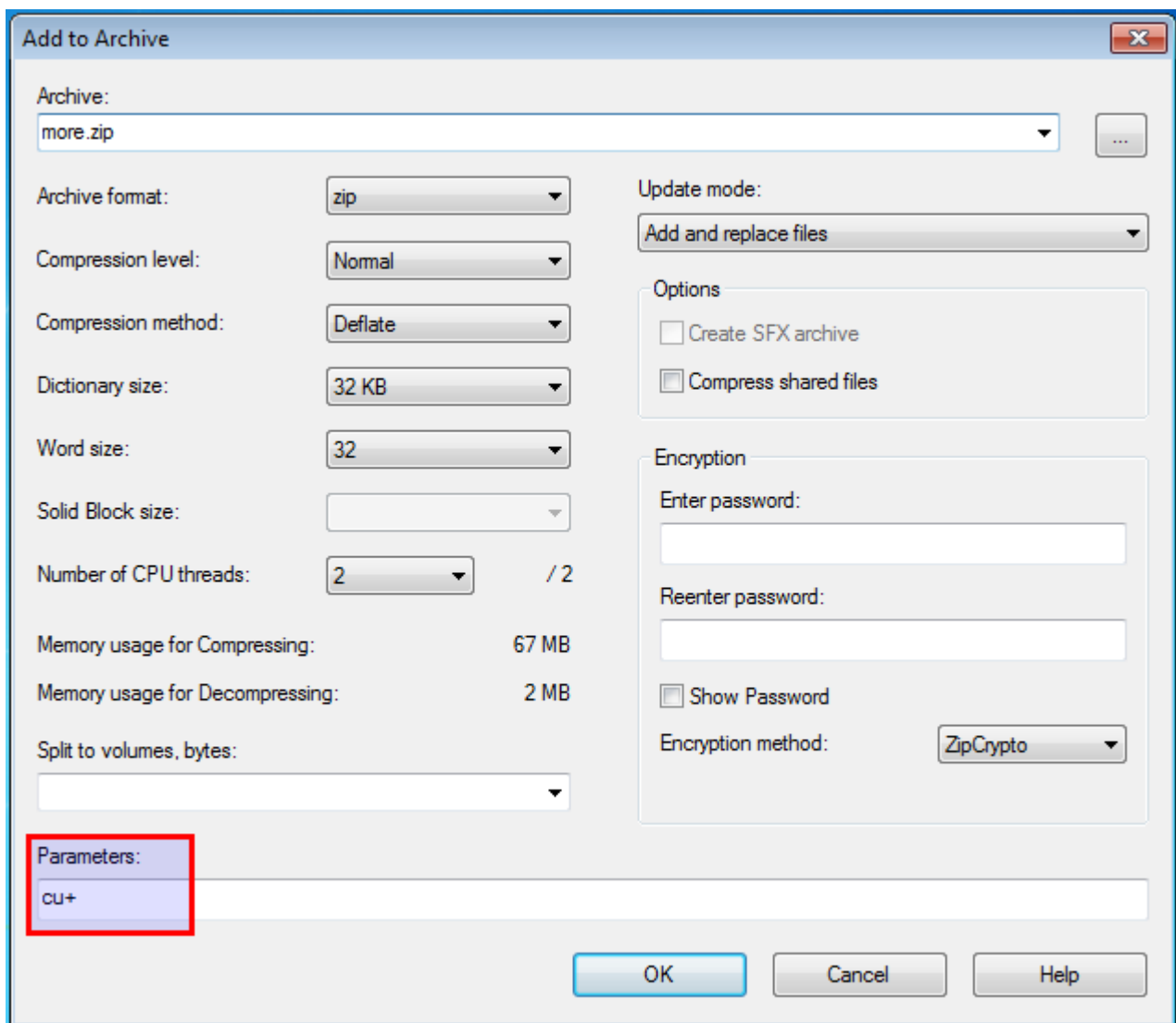
We've observated in some cases the execution with some restrictions goes right and not the case without any restriction.

Client configuration

ZIP filenames with wrong charset

When you download an OpenKM as ZIP maybe the included files have a wrong charset. This only happens with non-latin languages like Greek, Chinese or Arabic. This issue is corrected since OpenKM 4.1 and exported ZIP archives should be opened without problems from WinZip 14 or higher. But UNICODE support in ZIP is a little tricky and may cause problems. To ensure you won't have problems, try using 7-Zip which seems to handle these UNICODE filenames.

The other problem you may have is when you want to import a ZIP into OpenKM. In this case you can also get wrong filenames. Filenames should be encoded in UNICODE, and to ensure it you can force 7-Zip to use this encoding when creating a ZIP:



If you are on Windows you can download 7-Zip from <http://www.7-zip.org/>. You can also install the port of the 7-Zip compression tool for Linux:

```
$ sudo aptitude install p7zip-full p7zip
```

Microsoft Office Addin disappears after Office restart

Could be a problem with default template (normal.dot). To solve try it:

- *Open normal.dot*
- *Go to menu tools / personalize / Tool bars*
- *Open OpenKM toolbar*
- *Save normal.dot*

Error in Uploader / Scanner applet

To provide more complete information in order to determine the issue, please activate the Java Console. Read [How do I enable and view the Java Console?](#) for more info.

Browser configuration



On latest browsers versions the steps described below could not be exactly the same. Take it as a reference of what you should be looking for.

Firefox cause error when click on "Copy to clipboard"

To enable JavaScript "Copy to clipboard" buttons on Firefox and Mozilla, enter the following line into browser address bar: **"about:config"**.

The list of preferences should open in browser window. **Right-click somewhere** in the active part of preference list window and choose **New > Boolean** option from context menu. Paste the following Preference name into the New boolean value window input box: **"signed.applets.codebase_principal_support"**.

Unable to run Scanner applet under Linux

If you enable the Java Console and see this error:

```
java.lang.UnsatisfiedLinkError: uk.co.mmscomputing.util.JarLib.loadX: Could not find
```

You have to install the **libsane-dev** package:

```
$ sudo aptitude install libsane-dev
```

Internet Explorer error "Document name is different"

This is a known error in Internet Explorer family. To disable name mismatch check, change the value of configuration parameter named **"system.document.name.mismatch.check"** to false.



More information at [Other configuration parameters](#)

Internet Explorer error downloading files in HTTPS / SSL connections

There's a very old bug from version 5.5 until now that seems like Microsoft has not decided to fix, that causes a problem downloading files with IE behind SSL protocol (the force may be with you, hope some day they will decide to fix it).

The problem is caused by **"Cache-control: no-cache HTTP header to the response message"**, obviously that's valid for any browser except IE family ... other Microsoft IE hidden feature ?

Can see some supporting and forum information about it:

- [IE SSL download problem](#)

- <http://support.microsoft.com/kb/815313>
- <http://support.microsoft.com/kb/323308>
- <http://support.microsoft.com/kb/316431>

The solution is to change a couple of entries in the Windows Registry:

- Create a file called `ss_download_patch.reg`
- Copy the contents

```
REGEDIT4
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Internet Settings]
"DisableCachingOfSSLPages"=dword:00000000
"BypassSSLNoCacheCheck"=dword:00000001
"BypassHTTPNoCacheCheck"=dword:00000001
"SSLPragmaNoCache"=dword:00000001
```

- Execute it (will be imported into Windows Registry)
- Then you need to restart Windows for this to take effect.

Google Chrome and Java applets

As you can read at [Java and Google Chrome Browser](#), Chrome v45 no longer supports NPAPI which is required for Java applets. So you won't be able of using some interesting features in OpenKM like **Live Edit** o **Bulk Uploader** (both applets).

There are a couple of workarounds:

- Use another browser like Firefox, which still has NPAPI support.
- Go back to Chrome v44 and enable NPAPI support.

If you prefer keep using Chrome (option 2), follow these steps:

- Uninstall Google Chrome from your computer (**Control Panel** > **Uninstall a program**)
- Download [Google Chrome v44 portable](#) and install it.
- Once installed, use this URL to access browser configuration: <chrome://flags>
- Look for **Enable NPAPI**, press **Enable** link and restart Chrome.
- Now you would be able to use Java applets again.

Best practices reporting issues

To quickly help you in error identification and resolution, please try to follow the guideline below:

- Do a **detailed description** of the issue, that will help us understanding the problem do you have.
- When possible **make screenshots of all the steps you followed**. The idea is being able to reproduce the error or at least understanding better the context where has occurred.
- Videos are also welcome.
- Share the stack trace error from openkm.log file



Because support website have restriction in document size, share with us the download link with online file sharing tools like:

- <https://transfer.sh/>
- <https://filepush.co/>
- <https://filetransfer.io/>
- <https://justbeamit.com/>

Getting the stack trace

Linux

The **openkm.log** file is into the **\$TOMCAT/logs** folder. For example, in Linux a common location might be **"/home/openkm/tomcat-8.5.34/logs"**.

To watch how openkm.log changes in real time you can use the command line:

```
/home/openkm/tomcat-8.5.34/logs:$ tail -f openkm.log
```

To show the last 500 lines of the openkm.log file you can use the command line:

```
/home/openkm/tomcat-8.5.34/logs:$ tail -n 500 openkm.log
```

Windows

The **openkm.log** file is into the **\$TOMCAT/logs** folder. For example, in Linux a common location might be **"c:\tomcat-8.5.34\logs"**.

To watch how openkm.log changes in real time you can a windows tools like [mtail](#).



You have a copy of mtail tools into **c:\tomcat-8.5.34\extras**

Sample of stack trace capture



When you capture the stack trace error it is a good practice include some extra lines before and after the error happens.


```
2019-08-04 12:05:00,020 [Task Scheduler 18] [] INFO c.openkm.plugin.cron.BaseCronPlu
2019-08-04 12:05:00,032 [Task Scheduler 19] [] INFO c.openkm.plugin.cron.BaseCronPlu
2019-08-04 12:09:36,471 [http-nio-0.0.0.0-8080-exec-6] [okmAdmin] INFO com.openkm.ut
2019-08-04 12:09:36,603 [http-nio-0.0.0.0-8080-exec-6] [okmAdmin] ERROR com.openkm.ut
2019-08-04 12:09:36,612 [http-nio-0.0.0.0-8080-exec-6] [okmAdmin] ERROR c.o.s.admin.P
com.openkm.core.ParseException: Element type "hpanel" must be declared.
    at com.openkm.util.FormUtils.parsePropertyGroupsDefinition(FormUtils.java:308)
    at com.openkm.module.db.DbRepositoryModule.registerPropertyGroups(DbRepositoryModule
    at com.openkm.module.db.DbRepositoryModule.registerPropertyGroups(DbRepositoryModule
    at com.openkm.servlet.admin.PropertyGroupsServlet.doPost(PropertyGroupsServlet.java:
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:661)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:742)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilter
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.j
    at org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilter
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.j
    at com.openkm.servlet.UserLoggingFilter.doFilter(UserLoggingFilter.java:33)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilter
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.j
    at org.springframework.boot.web.filter.ApplicationContextHeaderFilter.doFilterIntern
    at org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilter
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.j
    at org.springframework.boot.actuate.trace.WebRequestTraceFilter.doFilterInternal(Web
    at org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilter
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.j
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(Fil
    at org.springframework.security.web.access.intercept.FilterSecurityInterceptor.invok
    at org.springframework.security.web.access.intercept.FilterSecurityInterceptor.doFil
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(Fil
    at org.springframework.security.web.access.ExceptionTranslationFilter.doFilter(Excep
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(Fil
    at org.springframework.security.web.session.SessionManagementFilter.doFilter(Sessio
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(Fil
    at org.springframework.security.web.authentication.AnonymousAuthenticationFilter.doF
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(Fil
    at org.springframework.security.web.servletapi.SecurityContextHolderAwareRequestFilt
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(Fil
    at org.springframework.security.web.savedrequest.RequestCacheAwareFilter.doFilter(Re
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(Fil
    at org.springframework.security.web.authentication.AbstractAuthenticationProcessingF
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(Fil
    at org.springframework.security.web.authentication.logout.LogoutFilter.doFilter(Log
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(Fil
    at org.springframework.web.filter.CorsFilter.doFilterInternal(CorsFilter.java:96)
    at org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(Fil
    at org.springframework.security.web.header.HeaderWriterFilter.doFilterInternal(Heade
    at org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(Fil
    at org.springframework.security.web.context.SecurityContextPersistenceFilter.doFilt
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(Fil
    at org.springframework.security.web.context.request.async.WebAsyncManagerIntegration
    at org.springframework.security.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(Fil
    at org.springframework.security.web.FilterChainProxy.doFilterInternal(FilterChainPro
```



```
at org.springframework.security.web.FilterChainProxy.doFilter(FilterChainProxy.java:
at org.springframework.web.filter.DelegatingFilterProxy.invokeDelegate(DelegatingFil
at org.springframework.web.filter.DelegatingFilterProxy.doFilter(DelegatingFilterPro
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilter
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.j
at org.springframework.web.filter.RequestContextFilter.doFilterInternal(RequestConte
at org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilter
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.j
at org.springframework.web.filter.HiddenHttpMethodFilter.doFilterInternal(HiddenHttp
at org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilter
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.j
at org.springframework.boot.actuate.autoconfigure.MetricsFilter.doFilterInternal(Met
at org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilter
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.j
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:19
at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:96
at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java
at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:140)
at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:80)
at org.apache.catalina.valves.AbstractAccessLogValve.invoke(AbstractAccessLogValve.j
at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:87)
at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:342)
at org.apache.coyote.http11.Http11Processor.service(Http11Processor.java:799)
at org.apache.coyote.AbstractProcessorLight.process(AbstractProcessorLight.java:66)
at org.apache.coyote.AbstractProtocol$ConnectionHandler.process(AbstractProtocol.jav
at org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:145
at org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:49)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61
at java.lang.Thread.run(Thread.java:748)
2019-08-04 12:10:00,000 [Task Scheduler 18] [] INFO c.openkm.plugin.cron.BaseCronPlu
2019-08-04 12:10:00,000 [Task Scheduler 6] [] INFO c.openkm.plugin.cron.BaseCronPlu
2019-08-04 12:10:00,000 [Task Scheduler 25] [] INFO c.openkm.plugin.cron.BaseCronPlu
2019-08-04 12:10:00,014 [Task Scheduler 6] [] INFO c.openkm.plugin.cron.BaseCronPlu
```

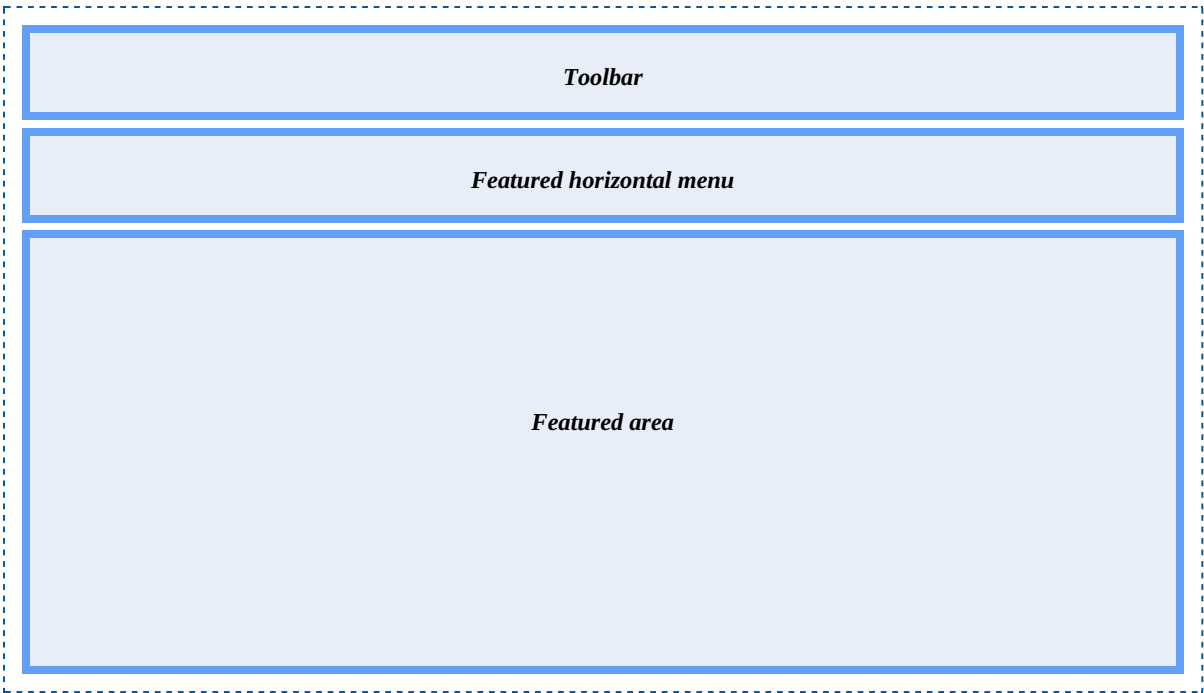
Administration guide

OpenKM provides easy access to the application configuration by means of the "Administration Tab" located on the administration system area. Relevant and useful information can be accessed to assist the administrator in controlling and configuring the OpenKM repository.




















Only users with **ROLE_ADMIN** can access the administration area. A new tab **Administration** will appear to these users at the right top corner of system toolbar.








Administration layout:



OpenKM administration provides access to:

Icon	Name	Description
	Home	Administration landing page. Here is shown: <ul style="list-style-type: none">• Application version.• Installation UUID.
	Configuration parameters	Almost all configuration parameters are set here. <div> With multi-tenant module enabled is possible to set different properties depending on the tenant.</div>

		 <i>Changes done here affect all the application.</i>
	Mime types	All registered document MIME types which OpenKM understands. You can modify, remove or add a new one.
	Statistics	Shows information about repository size . Also can check the evolution of the Pending task queue and Pending extraction queue .
	Scripting	Execute JAVA scripts.
	Metadata	Creates, modifies or removes metadata definitions.
	Users	Creates, modifies or removes users and roles.  <i>When the application is configured to connect LDAP, these values can only be changed on LDAP server because OpenKM connect to LDAP only in read only mode.</i>
	Profile	A profile is a specific set of configurations for the user interface.
	Database query	Execute queries to the database.
	Reports	Adds, removes or executes application reports.
	Activity log	User actions are registered in the OpenKM activity log.  <i>All activity log data is stored in OKM_ACTIVITY table.</i>
	Workflow	Adds, removes or view status of registered workflows.
	Automation	Sets automatic tasks based on application events like DOCUMENT_CREATION , among others.

	Crontab	<i>Adds or removes scheduled crontab tasks. The application uses these tasks to perform actions like indexing documents, antivirus analyse among others.</i>
	OMR (Object Mark Recognition)	<i>Sets Object Mark Recognition parameters.</i>
	Thesaurus	<i>Execute thesaurus build action.</i>
	Import	<i>Repository import.</i>
	Export	<i>Repository export</i>
	Languages	<i>Add, modifies or remove languages and manage translations.</i>
	Utilities	<p><i>A collection of advanced utilities:</i></p> <ul style="list-style-type: none"> • <i>Cache Stats.</i> • <i>Check e-mail.</i> • <i>Check text extraction.</i> • <i>Hibernate stats.</i> • <i>List indexes.</i> • <i>LogCat</i> • <i>Profile stats.</i> • <i>Purge permissions.</i> • <i>Rebuild indexes.</i> • <i>Repository checker.</i> • <i>Repository view.</i> • <i>System properties.</i> • <i>Css.</i>

Configuration parameters

OpenKM is preconfigured with a number of parameters. However, if you plan to use it in production you should take a look at [Recommended configuration parameters](#). To access Configuration parameters go to **Administration > Configuration parameters**.

Configuration parameters are shown in a table with three columns:

Property Key	Type	Value
--------------	------	-------


Property key is the unique identifier for each configuration property.

There are several configuration types:

- String.
- Integer (only **numeric** values are allowed).
- Boolean (only **true** or **false** values allowed).
- List (each break line is considered as new list element).
- Text.
- File.

To create your own parameters read [Creating your own parameters](#).

Edit a parameter

- Click **edit icon** .
- View and **modify the value**.
- Click **Edit button** to confirm the change.

Delete a parameter



You should not delete parameters if you're not totally sure about the implications of this action.

- Click **remove icon** .
- In **Delete confirmation view** click **Delete button**.

Backup configuration parameters


- Click at top right the **Configuration parameters export icon** .

- Browser will download an SQL file with the configuration.

Restore configuration parameters



Restoring configuration parameters will remove actual parameters and will be replaced by newers.

- Go to **bottom page**.
- Select **configuration parameter SQL file**.
- Click  **Import button**.

Check configuration

This action will check if some conflictive parameters are well configured (pdf2swf, convert tool, OCR engine, OpenOffice or LibreOffice path).






Although a parameter may pass this check, it that not means it is 100% correct. Take it as a light check which cannot covers all cases but is good approximation to discard most problems.






- Click at top right the **Check icon** .
- It will show the **Configuration check view**. If some parameters are misconfigured or wrong, it will be shown in red.

Recommended configuration parameters

There are some recommended changes in the configuration:

Field / Property	Type	Description
application.url	String	<p>This parameter is used in mail notifications and the copy-to-clipboard feature among others.</p> <p>By default OpenKM sets the access URL to <code>http://localhost:8080/OpenKM/index.jsp</code>, but obviously you don't want to access your OpenKM installation only from localhost.</p>
notification.message.body	HTML	<p>This parameter is used in mail notifications body.</p> <pre>Message: \${notificationMessage}
 User: \${userId}
 <#list documentList as doc>Document: \${doc.path}
</#list></pre>
notification.message.subject	Text	<p>This parameter is used in mail notifications subject. Help users identify that these emails are coming from OpenKM.</p> <pre>OpenKM - NOTIFICATION</pre>
subscription.message.body	HTML	<p>This parameter is used in mail user subscription notification body.</p> <pre>Document: \${documentPath}
 User: \${userId}
Event: \${eventType}
 Comment: \${subscriptionComment}
</pre>
subscription.message.subject	Text	<p>This parameter is used in mail subscription notification subject. Help users identify that these emails are coming from OpenKM.</p> <pre>OpenKM - \${eventType} - \${documentPath}</pre>

max.file.size	String	<p>It indicates the maximum allowed file size which can be uploaded to OpenKM. Default value is 0 indicates unlimited. Use size units with KB, MB, GB or TB, otherwise value will be considered as bytes.</p> <div>200MB</div>
managed.text.extraction	Boolean	Enable or disable.
managed.text.extraction.batch	Integer	<p>It indicates the number of documents being processed on each indexing cycle.</p> <div>  <p>When parameter "managed.text.extraction.concurrent" is enabled, is a good practice to set multiple of "managed.text.extraction.pool.threads" value.</p> </div>
managed.text.extraction.concurrent	Boolean	Enable or disable concurrent text extraction concurrent threads.
managed.text.extraction.pool.size	Integer	How many threads are launched concurrently every time. This number should be smaller or equal to the number of CPUs in the server.
managed.text.extraction.pool.threads	Integer	<p>Number of concurrent threads that will be used for text extraction.</p> <div>  <p>If "managed.text.extraction.pool.size" is 8 and "managed.text.extraction.pool.threads" is set to 16 then all these 16 threads will be launched using this pool. If every task take the same time to complete, in theory every pool thread will execute two tasks. Of course this is not true in the real world.</p> </div> <div>  <p>It's a good practice to set this number to be less or equal than CPU cores you want to use for text extraction.</p> </div>
managed.text.extraction.pool.timeout	Integer	How many minutes should the worker wait for the text extraction pool to complete the tasks. After these minutes, the remaining pool threads are interrupted.

<i>system.imagemagick.convert</i>	<i>String</i>	<i>Path to ImageMagick convert tool.</i>
<i>system.ocr</i>	<i>String</i>	<i>Path to OCR text extractor engine.</i>
<i>system.swftools.pdf2swf</i>	<i>String</i>	<i>Path to pd2swf conversion tool.</i>
<i>system.openoffice.path</i>	<i>String</i>	<i>Path to OpenOffice or LibreOffice installation.</i>
<i>logo.login</i>	<i>File</i>	<p><i>Login page logo.</i></p> <div>  <i>It automatically tries to resize image to 316px × 74px (suggested background #F1F3F5).</i> </div>
<i>logo.mobile</i>	<i>File</i>	<p><i>Mobile login page logo.</i></p> <div>  <i>It automatically tries to resize image to 161px × 38px (suggested background #F1F3F5).</i> </div>
<i>logo.report</i>	<i>File</i>	<p><i>Default logo used in Jasper Reports.</i></p> <div>  <i>It automatically tries to resize image to 150px × 35px (suggested background #FFF000).</i> </div>
<i>logo.tiny</i>	<i>File</i>	<p><i>Small logo at top right corner in frontend UI.</i></p> <div>  <i>It automatically tries to resize image to 51px × 13px (suggested background transparent).</i> </div>
<i>logo.favicon</i>	<i>File</i>	<p><i>Small icon in left side of browser url.</i></p> <div>  <i>The uploaded file must be a valid icon (16px x 16px).</i> </div>

Preview parameters

In this section you can find the most common configuration parameters related to preview. Some of these commands are included in OpenKM by default, like **pdf2swf** or **ffmpeg** (binaries compiles for Linux and Windows) but others need to be installed by the system administrator.

Linux

Field / Property	Type	Description
system.imagemagick.convert	String	Path to ImageMagick convert tool. <code>/usr/bin/convert</code>
system.ocr	String	Path to OCR text extractor engine. <code>/usr/bin/tesseract \${fileIn} \${fileOut}</code>
system.swftools.pdf2swf	String	Path to pd2swf conversion tool. <code>/home/openkm/tomcat-7.0.27/bin/pdf2swf -f -T 9 -t -s storeallcharacters \${fileIn} -o \${fileOut}</code>
system.pdfimages	String	Path to pdfimages extractor tool. <code>/usr/bin/pdfimages -j -f \${firstPage} -l \${lastPage} \${fileIn} \${imageRoot}</code>
system.openoffice.path	String	Path to OpenOffice or LibreOffice installation path. <code>/usr/lib/libreoffice</code>
system.ghostscript	String	Path tot the ghossript conversion tool. <code>/usr/bin/gs</code>

Windows

Field / Property	Type	Description
system.imagemagick.convert	String	Path to image magick convert tool. <code></code>

		<code>C:\openkm\bin\convert.exe</code>
system.ocr	String	<p>Path to ocr text extractor engine.</p> <pre>C:\openkm\tools\Tesseract-OCR\tesseract.exe \${fileIn} \${fileOut}</pre>
system.swftools.pdf2swf	String	<p>Path to pd2swf conversion tool.</p> <pre>C:\openkm\bin\pdf2swf.exe -f -T 9 -t -s storeallcharacters \${fileIn} -o \${fileOut}</pre>
system.pdfimages	String	<p>Path to pdfimages extractor tool.</p> <pre>C:\openkm\tools\pdfimages -j -f \${firstPage} -l \${lastPage} \${fileIn} \${imageRoot}</pre>
system.openoffice.path	String	<p>Path to Open Office or Libre Office installation path.</p> <pre>C:\Program Files (x86)\OpenOffice 4</pre>
system.ghostscript	String	<p>Path tot the ghossript conversion tool.</p> <pre>C:\openkm\bin\gswin32c.exe</pre>

LDAP configuration parameters

The OpenKM LDAP integration is based on a couple of LDAP searches to retrieve data.

LDAP queries are composed by tree elements:

1. Attribute or object to retrieve.
2. Node base to scope the request.
3. Filtering options.

Sample query

The query to retrieve mail attribute from an object of type person with sAMAccountName attribute value with {0} - this parameter indicate the userId - from node cn=users,dc=company,dc=local and descendants.

principal.ldap.mail.attribute	mail
principal.ldap.mail.search.base	cn=users,dc=company,dc=local
principal.ldap.mail.search.filter	(&(objectclass=person)(sAMAccountName={0}))

LDAP queries

- **principal.ldap.user.*** are used to retrieve user list.
- **principal.ldap.username.*** are used to retrieve user name.
- **principal.ldap.mail.*** are used to retrieve user mail adress.
- **principal.ldap.role.*** are used to retrieve role list.
- **principal.ldap.roles.by.user.*** are used to retrieve the roles of a user.
- **principal.ldap.users.by.role.*** are used to retrieve users of a role.


For retrieving data, are injected arguments into option filters. For example, for getting the user mail, the application use parameter {0} to set argument value - userId - into filtering options.


Queries that use parameters to filtering:

- **principal.ldap.mail.search.filter**=(sAMAccountName={0}). Where {0} is attribute value retrieved from principal.ldap.user.attribute.
- **principal.ldap.roles.by.user.search.filter**=(&(objectclass=user)(sAMAccountName={0})). Where {0} is attribute value retrieved from principal.ldap.user.attribute.
- **principal.ldap.username.search.filter**=(sAMAccountName={0}). Where {0} is attribute value retrieved from principal.ldap.user.attribute.

- **`principal.ldap.users.by.role.search.filter`**=(`&(objectClass=group)(cn={0})`). Where `{0}` is attribute value retrieved from `principal.ldap.role.attribute`.

Parameters

Field / Property	Type	Description
<code>principal.adapter</code>	String	Class used to connect with LDAP.
<code>system.login.lowercase</code>	String	By default is empty. In case connecting to AD (Microsoft Active Directory) must be set to " true ", that force all users to be logged with lowercase user Id. The reason is OpenKM is case sensitive and Microsoft Active Directory not.
<code>principal.ldap.server</code>	String	LDAP server.
<code>principal.ldap.security.principal</code>	String	LDAP user distinguished name (dn).
<code>principal.ldap.security.credentials</code>	String	LDAP user password.
		<p>This property specifies how the referrals sent by AD in the search results are handled by OpenKM. In almost most cases this value is empty.</p> <div data-bbox="762 1243 1404 1910" data-label="Complex-Block">  <p>Note for Active Directory (AD) users: AD servers are apparently unable to handle referrals automatically, which causes a <code>PartialResultException</code> to be thrown whenever a referral is encountered in a search. To avoid this, set the <code>ignorePartialResultException</code> property to <code>true</code>. There is currently no way of manually handling these referrals in the form of <code>ReferralException</code>, i.e. either you get the exception (and your results are lost) or all referrals are ignored (if the server is unable to handle them properly. Neither is there any simple way to get notified that a <code>PartialResultException</code> has been ignored (other than in the log).</p> <p>More information at Spring LdapTemplate.</p> </div> <p>Values might be:</p>

<i>principal.ldap.referral</i>	String	<p>"ignore"</p> <p>If <code>principal.ldap.referral=ignore</code>, the following exception will be thrown when a referral is encountered:</p> <div style="border: 1px dashed gray; padding: 5px; margin: 10px 0;"> <pre>javax.naming.PartialResultException: Unprocessed Continuation Reference(s); remaining name 'dc=company,dc=com'</pre> </div> <p>"follow"</p> <p>If <code>principal.ldap.referral=follow</code>, OpenKM will automatically follow the referral. To be successful, make sure OpenKM can access the referred server. The errors about <code>ReferralExceptions</code> should not be shown.</p> <p>"throw"</p> <p>If <code>principal.ldap.referral=throw</code>, the following exception will be thrown:</p> <div style="border: 1px dashed gray; padding: 5px; margin: 10px 0;"> <pre>com.sun.jndi.ldap.LdapReferralException: Continuation Reference; remaining name 'dc=company,dc=com'</pre> </div> <div style="border: 1px dashed blue; padding: 5px; margin: 10px 0; background-color: #e6f2ff;">  More information at Referrals in the JNDI. </div>
<i>principal.ldap.users.from.roles</i>	Boolean	When "true" tries to retrieve all users from roles list.
<i>principal.ldap.user.attribute</i>	String	User attribute.
<i>principal.ldap.user.search.base</i>	List	List of nodes base.
<i>principal.ldap.user.search.filter</i>	String	Filter options.
<i>principal.ldap.username.attribute</i>	String	Username attribute
<i>principal.ldap.username.search.base</i>	String	Node base.

<i>principal.ldap.username.search.filter</i>	String	<i>Filter options.</i>
<i>principal.ldap.mail.attribute</i>	String	<i>Mail attribute.</i>
<i>principal.ldap.mail.search.base</i>	String	<i>Node base.</i>
<i>principal.ldap.mail.search.filter</i>	String	<i>Filter options.</i>
<i>principal.ldap.role.attribute</i>	String	<i>Role attribute.</i>
<i>principal.ldap.role.search.base</i>	List	<i>List of nodes base.</i>
<i>principal.ldap.role.search.filter</i>	String	<i>Filter options.</i>
<i>principal.ldap.roles.by.user.attribute</i>	String	<i>User attribute.</i>
<i>principal.ldap.roles.by.user.search.base</i>	String	<i>Node Base.</i>
<i>principal.ldap.roles.by.user.search.filter</i>	String	<i>Filter options.</i>
<i>principal.ldap.users.by.role.attribute</i>	String	<i>Role attribute.</i>
<i>principal.ldap.users.by.role.search.base</i>	String	<i>Node base.</i>
<i>principal.ldap.users.by.role.search.filter</i>	String	<i>Filter options.</i>

Sample configuration





<i>Field / Property</i>	<i>Type</i>	<i>Description</i>
<i>principal.adapter</i>	String	<code>com.openkm.principal.LdapPrincipalAdapter</code>
<i>system.login.lowercase</i>	String	<code>true</code>
<i>principal.ldap.server</i>	String	<code>ldap://192.168.1.20:389</code>

<i>principal.ldap.security.principal</i>	<i>String</i>	<i>CN=Administrator,OU=OpenKM,DC=company,DC=com</i>
<i>principal.ldap.security.credentials</i>	<i>String</i>	<i>password</i>
<i>principal.ldap.referral</i>	<i>String</i>	
<i>principal.ldap.users.from.roles</i>	<i>Boolean</i>	<i>false</i>
<i>principal.ldap.user.attribute</i>	<i>String</i>	<i>sAMAccountName</i>
<i>principal.ldap.user.search.base</i>	<i>List</i>	<i>DC=company,DC=com</i>
<i>principal.ldap.user.search.filter</i>	<i>String</i>	<i>objectclass=person</i>
<i>principal.ldap.username.attribute</i>	<i>String</i>	<i>cn</i>
<i>principal.ldap.username.search.base</i>	<i>String</i>	<i>DC=company,DC=com</i>
<i>principal.ldap.username.search.filter</i>	<i>String</i>	<i>(sAMAccountName={0})</i>
<i>principal.ldap.mail.attribute</i>	<i>String</i>	<i>mail</i>
<i>principal.ldap.mail.search.base</i>	<i>String</i>	<i>DC=company,DC=com</i>
<i>principal.ldap.mail.search.filter</i>	<i>String</i>	<i>(sAMAccountName={0})</i>
<i>principal.ldap.role.attribute</i>	<i>String</i>	<i>cn</i>
<i>principal.ldap.role.search.base</i>	<i>List</i>	<i>DC=company,DC=com</i>
<i>principal.ldap.role.search.filter</i>	<i>String</i>	<i>objectclass=group</i>
<i>principal.ldap.roles.by.user.attribute</i>	<i>String</i>	<i>memberOf</i>
<i>principal.ldap.roles.by.user.search.base</i>	<i>String</i>	<i>DC=company,DC=com</i>

<i>principal.ldap.roles.by.user.search.filter</i>	<i>String</i>	<code>(&(objectclass=person)(sAMAccountName={0}))</code>
<i>principal.ldap.users.by.role.attribute</i>	<i>String</i>	<code>member</code>
<i>principal.ldap.users.by.role.search.base</i>	<i>String</i>	<code>DC=company,DC=com</code>
<i>principal.ldap.users.by.role.search.filter</i>	<i>String</i>	<code>(&(objectClass=group)(cn={0}))</code>

User interface configuration parameters

The uploaded images - except in case of the favicon - are automatically resized to match the right size. Most image formats like jpg, png, etc are allowed .

Field / Property	Type	Description
default.lang	String	Forces user interface language to a specific language. By default this parameter is empty. The value is based on international language code. <div>en-GB</div>
security.mode.multiple	Boolean	Enabled alternative security mode that allows to apply several changes at same time. Only one change is applied each time. By default the value is "false". <div> After the parameter is changed, refresh the browser to take effect.</div> <div>false</div>
logo.favicon	String	Website icon.
logo.login	String	Login page logo. <div> Automatically it tries to resize image to 316px × 74px (suggested background #F1F3F5).</div>
logo.mobile	String	Mobile logo. <div> Automatically it tries to resize image to 161px × 38px (suggested background #F1F3F5).</div>
logo.report	String	Reports logo. <div> Automatically it tries to resize image to 150px × 35px (suggested background #FFF000).</div>

logo.tiny	String	A small logo is used at the top right in OpenKM user interface.
text.banner	String	<p>Text shown at top of the logo in the landing page.</p> <div>  Automatically it tries to resize image to 51px x 13px (suggested background #E5E5E1). </div>
text.title	String	<p>Browser title. This text is set between</p> <div> <code>\${text.title}</code> </div> <p>html tags.</p>
text.welcome	String	<p>Text shown in the login table at the landing page.</p> <div> <p>Welcome to OpenKM !</p> <p>Use a valid username and password to access to OpenKM user Desktop.</p> </div>
keep.session.alive.interval	Integer	<p>Some administration tasks take a long time, like repository checker and rebuild Lucene indexes, these can cause the user session to be lost. For this reason these type of operations are forced to maintain the session alive. By default is set to 5 minutes.</p> <div> 5 </div>
schedule.session.keepalive	Integer	<p>It is used by the user interface to maintain the browser connected to OpenKM even when a user stays for some time without performing any operation. For this reason, it should never be lower than server session timeout (by default set to 15 minutes). By default is set to 5 minutes.</p> <div> 5 </div>
schedule.dashboard.refresh	Integer	<p>The application periodically does some operations like refreshing the dashboard panel to get new imported mails, getting repository news or showing latest documents created by other users among others. The time set the period between dashboard panel will be refresh.</p> <div> 15 </div>

<i>schedule.ui.notification</i>	<i>Integer</i>	<p>User receives notification with a popup. The time - in minutes - indicate the period used by application looking for new messages in server.</p> <div>1</div>
<i>system.previewer</i>	<i>Select</i>	<p>The application can use "flexpaper" or "zviewer" tool to preview documents. By default flexpaper previewer is enabled.</p> <div>flexpaper</div>
<i>min.search.characters</i>	<i>Integer</i>	<p>Sets the minimum characters required to execute a search. With less characters the search button will not be enabled.</p> <div>3</div>


Workflow configuration parameters

Explain for what these parameters are:

Field / Property	Type	Description
workflow.run.config.form	String	<p>The form name in case the workflow needs parameters at the start. By default this form is called "run_config".</p> <pre>run_config</pre>
workflow.start.task.auto.run	String	<p>If OpenKM should run the task defined in the start node. This start dummy task is usually used to assign the "initiator" swimlane. By default the value is set to "true". You don't need to modify this value.</p> <pre>true</pre>

Notification configuration parameters

When a user is subscribed to a document, any change in the document will be notified to by email. These email messages are created using some defined templates. These default templates to create messages can be adapted to your own style.

Field / Property	Type	Description
send.mail.from.user	Boolean	<p>By default the application sends mail notifications setting the from mail field value as the user mail address. When this parameter is disabled it will use the default from mail address for all the mails sent by the application (all mails sent by the application will have the same from mail address).</p> <div>  <p>When mail server have restricted to change the default from mail field (for example when is not allowed to use general@mail.com account to send a mail with from mail field value user@mail.com , usually this kind of mail server configuration is set to prevent spam) then this parameter must be set as "false".</p> </div> <div>true</div>
notification.message.subject	Text	<p>Subject of the message sent when a user has chosen notify something to other.</p> <div>OpenKM - NOTIFICATION</div>
notification.message.body	HTML	<p>Body of the message sent when a user has chosen notify something to other.</p> <div> <pre> Message: \${notificationMessage}
 User: \${userId}
 <#list documentList as doc>Document: \${doc.path}
</#list> </pre> </div>
subscription.message.subject	Text	<p>Subject of the message sent automatically caused by some event. That happens because the user is subscribed to some node.</p>

		<div>OpenKM - \${eventType} - \${documentPath}</div>
subscription.message.body	HTML	<p>Body of the message sent automatically caused by some event. That happens because user is subscribed to some node.</p> <div> Document: \${documentPath}
 User: \${userId}
Event: \${eventType}
 Comment: \${subscriptionComment}
 </div>
proposed.subscription.message.subject	Text	<p>Subject of the message sent when a user propose some node subscription to other.</p> <div>OpenKM - PROPOSED SUBSCRIPTION</div>
proposed.subscription.message.body	HTML	<p>Body of the message sent when a user propose some node subscription to other.</p> <div> Comment: \${proposedSubscriptionComment}
 User: \${userId}
 <#list documentList as doc>Document: \${doc.path}
</#list> </div>
notify.twitter.status	Text	<p>This is the text used in the notification twitter message.</p> <div> OpenKM - \${documentUrl} - \${documentPath} - \${userId} - \${eventType} Edit Delete </div>
notify.twitter.user	String	Twitter account username.
notify.twitter.password	String	Twitter account password.

Substitutions

For **notification.message.body** property:

- **\${documentUrl}** or **\${docUrl}** shows the node URL..
- **\${documentPath}** or **\${docPath}** shows the node path.
- **\${documentName}** or **\${docName}** show the node name.
- **\${userId}** show the user id who caused the action.
- **\${notificationMessage}** show the notification message.

For **subscription.message.body**, **proposed.subscription.message.body** and **notify.twitter.status** properties:

- **\${documentUrl}** or **\${docUrl}** show the node URL.
- **\${documentPath}** or **\${docPath}** show the node path.
- **\${documentName}** or **\${docName}** show the node name.
- **\${userId}** show the user id who caused the event.
- **\${eventType}** show the event type.
- **\${subscriptionComment}** show the subscription comment.



FreeMarker does not allow you to render null values (will throw an exception). If you expect a value to be null you can also use the ! operator: **\${content.title!}**. This is the equivalent to **\${content.title!}**.



More information at:

- [java.text.MessageFormat](#)

Document version numeration configuration

We support different numeration schemas. By default OpenKM uses

`com.openkm.vernum.MajorMinorVersionNumerationAdapter` which is similar to Jackrabbit implementation.



More information about how to [Creating your own Version Number Adapter](#).

Field / Property	Type	Description
<code>version.numeration.adapter</code>	String	<p>Numeration schema.</p> <p>Available schemas :</p> <ul style="list-style-type: none"> <code>com.openkm.vernum.PlainVersionNumerationAdapter</code> <code>com.openkm.vernum.MajorMinorVersionNumerationAdapter</code> <code>com.openkm.vernum.BranchVersionNumerationAdapter</code> <code>com.openkm.vernum.LetterVersionAdapter</code> <code>com.openkm.vernum.MajorMinorReleaseVersionNumerationAdapter</code> <p><code>com.openkm.vernum.MajorMinorVersionNumerationAdapter</code></p>
<code>version.numeration.format</code>	String	<p>Numeration number format. Used in <code>PlainVersionNumerationAdapter</code>.</p> <p><code>%d</code></p>

Some Numeration Adapters like **`PlainVersionNumerationAdapter`** may have additional configuration parameters like **`version.numeration.format`** which is used to define the numeration number format. By default this is `"%d"` but you can change to `"%02d"` to create versions like "01", "02", etc.



Restarting application is needed after changing the value of **`version.numeration.adapter`**.

Also note that if already there are documents in the repository, changing the Numeration Adapter may cause problems because they are not compatible between them.

Version numeration schemas

Numeration adapter	First version	Next versions
<code>com.openkm.vernum.PlainVersionNumerationAdapter</code>	1	2, 3, 4, etc.
<code>com.openkm.vernum.MajorMinorVersionNumerationAdapter</code>	1.0	1.1, 1.2, 1.3, etc.

<i>com.openkm.vernum.BranchVersionNumerationAdapter</i>	1	2, 3, 2.1, 2.2, etc.
<i>com.openkm.vernum.LetterVersionAdapter</i>	A	A,B,C, etc.
<i>com.openkm.vernum.MajorMinorReleaseVersionNumerationAdapter</i>	1.0.0	1.1.0, 1.2.0, 1.2.1, etc.

Migration between versions



We recommend you set the correct numeration at the beginning of using the application . Otherwise will be needed a migration of existing numeration and this operation is not always easy.

You can change the default **Numeration Adapter** using the "**version.numeration.adapter**" configuration parameter. You need to restart OpenKM every time you change this parameter. Also note that if already there are documents in the repository, changing the **Numeration Adapter** may cause problems when they are not compatible between them.

If you change the version numeration adapter you also need to update the existing numeration, for example the migration from **PlainVersionNumerationAdapter** to **MajorMinorVersionNumerationAdapter**.

```
-- HQL queries
UPDATE NodeDocumentVersion ndv SET ndv.name = '01' WHERE ndv.name='1.0';
UPDATE NodeDocumentVersion ndv SET ndv.name = '02' WHERE ndv.name='1.1';
UPDATE NodeDocumentVersion ndv SET ndv.name = '03' WHERE ndv.name='1.2';
```

Password validation configuration

Validation password enables the application to set a couple of rules that should be accomplished by passwords. By default, application makes no password checking (**validator.password** property is set by default to **com.openkm.validator.password.NoPasswordValidator**).

To enable validator password must change to **com.openkm.validator.password.CompletePasswordValidator** or [Creating your own password validator](#).

Field / Property	Type	Description
validator.password	String	<p>Description</p> <p>Available:</p> <ul style="list-style-type: none"> <code>com.openkm.validator.password.NoPasswordValidator</code> <code>com.openkm.validator.password.CompletePasswordValidator</code> <p><code>com.openkm.validator.password.NoPasswordValidator</code></p>
validator.password.min.length	String	<p>Minimum length allowed.</p> <p>Numeric value greater than 0.</p>
validator.password.max.length	String	<p>Maximum length allowed.</p> <p>Numeric value greater than 0.</p>
validator.password.min.lowercase	String	<p>Minimum number of lower-case characters.</p> <p>Numeric value greater than 0.</p>
validator.password.min.uppercase	String	<p>Minimum number of upper-case characters.</p> <p>Numeric value greater than 0.</p>
validator.password.min.digits	String	<p>Minimum number of digits.</p> <p>Numeric value greater than 0.</p>
validator.password.mini.special	String	<p>Minimum number of special characters.</p>

		Numeric value greater than 0.
--	--	-------------------------------

Sample configuration

- Must have at least 8 characters.
- At least must have 1 upper-case character.
- At least must have 1 digit.

Field / Property	Type	Description
validator.password	String	<code>com.openkm.validator.password.CompletePasswordValid</code>
validator.password.min.length	String	8
validator.password.max.length	String	
validator.password.min.lowercase	String	
validator.password.min.uppercase	String	1
validator.password.min.digits	String	1
validator.password.mini.special	String	

Switch application to read-only mode

It can force the whole repository to be in read-only mode. This is useful when doing maintenance tasks, or while migrating and you wish to get the system alive for your users but want to ensure nobody can change anything.

Field / Property	Type	Description
<code>system.readonly</code>	Boolean	Switch whole repository to be read-only. By default is set to false. <div><code>false</code></div>





If you **restart the application**, this parameter will be **automatically set to true**.






For security and performance reasons, when you execute **Lucene rebuild indexes**, the system will **switch automatically to read-only mode** until the process is finished. Do not change manually, wait until be completed.

Security configuration parameters

Field / Property	Type	Description
default.user.role	String	<p>Sets the default general connection role.</p> <div>Any user to successfully login in the screen needs to have default.user.role or default.admin.role, otherwise will get 403 error.</div> <div>ROLE_USER</div>
default.admin.role	String	<p>Sets the default administration connection role. This role will get users access to Administration tools.</p> <div>  Any user to successfully login in the screen need to have default.user.role or default.admin.role, otherwise will get 403 error. </div> <div>ROLE_ADMIN</div>
user.assign.document.creation	Boolean	<p>By default when a user creates a node, he is added to the node with full permissions. You can disable this behavior by setting this parameter to false. By default the parameters is set to true.</p> <div>true</div>
user.password.reset	Boolean	<p>Sometimes the user loses or forgets their password. When this option is enabled, used in landing page have got the option to restart their password and be notified with a newer one by mail. By default this option is set to false.</p> <div>false</div>
system.antivir	String	<p>Path to antivirus.</p> <div>  An antivirus can decrease performance, taking several seconds to analyze each document and consuming a lot of CPU resources. </div>

<i>security.access.manager</i>	<i>String</i>	<p>Sets security access manager determinate the way on how the security is evaluated. Default value is "simple".</p> <p>Values:</p> <ul style="list-style-type: none"> • <i>simple</i>. • <i>recursive</i>. • <i>read_recursive</i>. <div style="border: 1px solid black; background-color: #ffffcc; padding: 2px; margin: 10px 0;"><i>simple</i></div> <p>Takes a look at Security configuration</p>
<i>security.search.evaluation</i>	<i>String</i>	<p>Sets the security search manager and determinates the way how the security is evaluated by search engine.</p> <p>Values:</p> <ul style="list-style-type: none"> • <i>lucene</i>. • <i>am_more</i>, <i>am_window</i> or <i>am_limited</i>. <p>Take a look at Security configuration</p>
<i>security.extended.mask</i>	<i>Integer</i>	<p>Security evaluation can be extended to downloaded files, starts workflows, adds, removes or modifies property groups (metadata) or compact history. Default value is empty.</p> <p>Values:</p> <ul style="list-style-type: none"> • <i>DOWNLOAD</i> = 1024; • <i>START_WORKFLOW</i> = 2048; • <i>COMPACT_HISTORY</i> = 4096; • <i>PROPERTY_GROUP</i> = 8192; <p>For example to enable download and property groups you should put the mask 9216 (1024+8192)</p> <div style="border: 1px solid black; background-color: #ffffcc; padding: 2px; margin: 10px 0;"><i>9216</i></div> <p>Take a look at Security configuration</p>

<i>security.live.change.node.limit</i>	Integer	<p>When security changes affect more than the values set in this property the security changes are applied as a background task.</p> <div>100</div> <p>Take a look at Security configuration</p>
<i>system.login.lowercase</i>	Boolean	<p>By default is empty. In case connecting to AD (Microsoft Active Directory) must be set to "true", that force all users to be logged with lowercase user Id. The reason is OpenKM is case sensitive and Microsoft Active Directory not.</p>
<i>principal.adapter</i>	String	<div>  <p>You must restart OpenKM service after you change this parameter</p> </div> <p>OpenKM can handle user access using the Spring Security framework. OpenKM needs an available method for reading users and roles, so when users are stored in a database (as is the default), the class DatabasePrincipalAdapter does this job.</p> <div>com.openkm.core.DatabasePrincipalAdapter</div> <p>If you configure OpenKM to authenticate against an LDAP server, you need to configure another principal adapter like LdapPrincipalAdapter.</p> <div>  <p>More information at:</p> <ul style="list-style-type: none"> • LDAP configuration. • Creating your own login module. </div>
<i>principal.identifier.validation</i>	String	<p>Sets a regular expression to validate on the creation of user names and roles names.</p> <div>^[a-zA-Z0-9_]+\$</div> <div>  <p>This parameter only can be used in combination with "principal.adapter" parameter value "com.openkm.core.DatabasePrincipalAdapter".</p> </div>

Extensions configuration parameters

HTML Editor 3

The application comes with integrated WYSIWYG editor based on tinyMCE v3.

The components of the editor can be parametrized with these parameters:

Field / Property	Type	Description
<i>tinymce.plugins</i>	<i>String</i>	<p>Set enabled tinyMCE plugins.</p> <p>Values:</p> <ul style="list-style-type: none"> • <i>codehighlighting</i> • <i>autolink</i> • <i>lists</i> • <i>pagebreak</i> • <i>style</i> • <i>layer</i> • <i>table</i> • <i>save</i> • <i>advhr</i> • <i>advimage</i> • <i>advlink</i> • <i>emotions</i> • <i>iespell</i> • <i>inlinepopups</i> • <i>insertdatetime</i> • <i>preview</i> • <i>media</i> • <i>searchreplace</i> • <i>print</i> • <i>contextmenu</i> • <i>paste</i> • <i>directionality</i> • <i>fullscreen</i> • <i>noneditable</i>

		<ul style="list-style-type: none"> • <i>visualchars</i> • <i>nonbreaking</i> • <i>xhtmlxtras</i> • <i>template</i> • <i>advlist</i> <div> <i>codehighlighting, autolink, lists, pagebreak, style, layer, table, save, advhr, advimage, advlink, emotions, iespell, inlinepopups, insertdatetime, preview, media, searchreplace, print, contextmenu, paste, directionality, fullscreen, noneditable, visualchars, nonbreaking, xhtmlxtras, template, advlist</i> </div>
<i>tinymce.skin</i>	<i>String</i>	<p>Set the tinyMCE skin</p> <p>Values:</p> <ul style="list-style-type: none"> • <i>o2k7</i> <div> <i>o2k7</i> </div>
<i>tinymce.skin.variant</i>	<i>String</i>	<p>Set the tinyMCE skin variant</p> <p>Values:</p> <ul style="list-style-type: none"> • <i>silver</i> <div> <i>silver</i> </div>
<i>tinymce.theme</i>	<i>String</i>	<p>Set the tinyMCE theme</p> <p>Values:</p> <ul style="list-style-type: none"> • <i>simple</i> • <i>advanced</i> <div> <i>advanced</i> </div>
		<p>Set elements present in first toolbar.</p>

<p><i>tinymce.theme.buttons1</i></p>	<p>String</p>	<p>Values:</p> <ul style="list-style-type: none"> • <i>okm_checkin</i> • <i>okm_cancelcheckout</i> • <i>okm_searchDocument</i> • <i>okm_searchFolder</i> • <i>okm_searchImage</i> • <i>newdocument</i> • <i>codehighlighting</i> • <i>bold</i> • <i>italic</i> • <i>underline</i> • <i>strikethrough</i> • <i>justifyleft</i> • <i>justifycenter</i> • <i>justifyright</i> • <i>justifyfull</i> • <i>styleselect</i> • <i>formatselect</i> • <i>fontselect</i> • <i>fontsizeselect</i> <div style="border: 1px solid yellow; padding: 5px; margin-top: 10px;"> <p><i>okm_checkin, okm_cancelcheckout, okm_searchDocument, okm_searchFolder, okm_searchImage, newdocument, codehighlighting, , bold, italic, underline, strikethrough, , justifyleft, justifycenter, justifyright, justifyfull, styleselect, formatselect, fontselect, fontsizeselect</i></p> </div>
		<p><i>Set elements present in second toolbar.</i></p> <p>Values:</p> <ul style="list-style-type: none"> • <i>cutcopy</i> • <i>paste</i> • <i>pastetext</i> • <i>pasteword</i>

<i>tinymce.theme.buttons2</i>	<i>String</i>	<ul style="list-style-type: none"> • <i>search</i> • <i>replace</i> • <i>bullist</i> • <i>numlist</i> • <i>outdent</i> • <i>indent</i> • <i>blockquote</i> • <i>undo</i> • <i>redo</i> • <i>link</i> • <i>unlink</i> • <i>anchor</i> • <i>image</i> • <i>cleanup</i> • <i>help</i> • <i>code</i> • <i>insertdate</i> • <i>inserttime</i> • <i>preview</i> • <i>forecolor</i> • <i>backcolor</i> <div data-bbox="576 1301 1401 1547" style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <i>cut, copy, paste, pastetext, pasteword, , search, replace, , bullist, numlist, , outdent, indent, blockquote, , undo, redo, , link, unlink, anchor, image, cleanup, help, code, , insertdate, inserttime, preview, , forecolor, backcolor</i> </div>
		<p><i>Set elements present in third toolbar.</i></p> <p><i>Values:</i></p> <ul style="list-style-type: none"> • <i>tablecontrols</i> • <i>hr</i> • <i>removeformat</i> • <i>visualaid</i> • <i>sub</i>

<i>tinymce.theme.buttons3</i>	<i>String</i>	<ul style="list-style-type: none"> • <i>sup</i> • <i>charmap</i> • <i>emotions</i> • <i>iespell</i> • <i>media</i> • <i>advhr</i> • <i>print</i> • <i>ltr</i> • <i>rtl</i> <div> <i>tablecontrols, , hr, removeformat, visualaid, , sub, sup, , charmap, emotions, iespell, media, advhr, , print, , ltr, rtl</i> </div>
<i>tinymce.theme.buttons4</i>	<i>String</i>	<p><i>Set elements present in fourth toolbar.</i></p> <p><i>Values:</i></p> <ul style="list-style-type: none"> • <i>insertlayer</i> • <i>moveforward</i> • <i>movebackward</i> • <i>absolute</i> • <i>styleprops</i> • <i>cite</i> • <i>abbr</i> • <i>acronym</i> • <i>del</i> • <i>ins</i> • <i>attribs</i> • <i>visualchars</i> • <i>nonbreaking</i> • <i>template</i> • <i>pagebreak</i> <div> <i>insertlayer, moveforward, movebackward, absolute, , styleprops, , cite, abbr, acronym, del, ins, attribs, , visualchars, nonbreaking, template, pagebreak</i> </div>

DXF applet viewer

Application comes with DXF applet viewer. If you observe problems on previewer text, you should configure the URL to indicate an extra fonts repository used in AutoCAD files. Simply copy the files to some public URL that can be resolved by desktop computer.



The font file names should be in lower case (rename if you have upper-case font file names).

Field / Property	Type	Description
<code>dxf.applet.debug.window</code>	Boolean	Enable debug in java console view. By default is disabled. <code>false</code>
<code>dxf.applet.default.bigfont</code>	String	Font might be used by big text. By default is empty. <code>arial</code>
<code>dxf.applet.default.font</code>	String	Set the default font. By default is empty. <code>arial</code>
<code>dxf.applet.fontdir</code>	String	Public URL with extra font repository. By default is empty. <code>http://fonts.openkm.com</code>

Zoho

The Zoho Docs is a Web-based online office suite containing word processing, spreadsheets, etc. Integration make possible online edition on documents stored into OpenKM repository.

You can obtain the user and password registering at <http://zapi.zoho.com>. Once registered and logged, you can request the API Key and the Secret Key. In the Zoho Dashboard you can also check how many edition requests you have performed.

Field / Property	Type	Description
<code>zoho.api.key</code>	String	The api key.
<code>zoho.secret.key</code>	String	The secret key.
<code>zoho.user</code>	String	Zoho username.

zoho.password	String	Zoho password.
----------------------	---------------	----------------

OpenMeetings

OpenMeetings is a free browser-based software that allows you to set up instantly a conference in the Web. You can use your microphone or webcam, share documents on a white board, share your screen or record meetings. It is available as hosted service or you download and install a package on your server with no limitations in usage or users.

Field / Property	Type	Description
openmeetings.url	String	The url of the OpenMeetings server. <input type="text" value="localhost"/>
openmeetings.port	String	Default port is 5080. <input type="text" value="5080"/>
openmeetings.user	String	OpenMeetings user with administration grants.
openmeetings.credentials	String	OpenMeetings password.

DICOM preview


With DICOM preview can be shown medical images. DICOM preview allows manipulate images from anywhere as easy as surfing the Internet with zero training.

Field / Property	Type	Description
dicom.url	String	URL of the DICOM service.

Extratab extension

The ExtraTab allows to easily add a new tab with a name and specific URL.

Field / Property	Type	Description
extra.tab.workspace.label	String	Label of the tab. <input type="text" value="Some text"/>

<i>extra.tab.workspace.url</i>	<i>String</i>	<p>The URL of the tab.</p> <div><code>http://www.openkm.com</code></div> <div> <i>If URL does not start with http it's considered as OpenKM relative path and will be merged with your OpenKM server URL like YOUR_OPENKM_URL/url</i></div>
---------------------------------------	----------------------	---

Performance configuration parameters

Text extractor queue

There are several configuration parameters to adapt the document text extraction to your needs. Text extraction is one of the processes that can use a lot of hardware resources (memory and CPU's). Aggressive text extraction policy can decrease the performance of the application and affect the end user feeling with user interface.

When documents are uploaded are automatically added into **Text extraction queue**. Can see Text extraction queue at **Administration > Statistics > Text extraction queue**.

There's a crontab task named "**Text extractor worker**" at **Administration > Cron tab** what control the indexing cycle



When a document is uploaded, it will not be able to search by content until is processed by Text extraction queue.



In case the application need indexing a lot of files per day, or you imported a considerable amount of files, consider create two crontab task to change these parameters between day (less aggressive) and night (more aggressive).



Some mime-types like pdf or images (what needs OCR engine) can use a lot of CPU (top 100%).

Field / Property	Type	Description
managed.text.extraction.batch	Boolean	<p>Indicate the number of documents to be processed on each indexing cycle.</p> <div> When parameter "text.extraction.concurrent" is enabled, is a good practice to be multiple of "text.extraction.threads" value. </div> <p>20</p>
managed.text.extraction.concurrent	Boolean	<p>Enable or disable concurrent text extraction concurrent threads feature.</p> <p>True</p>
managed.text.extraction.threads	Integer	<p>Number of concurrent threads will be used for text extraction feature.</p> <div> This number should be less or equal the number of </div>

		hardware cores.
		2

OpenOffice LibreOffice conversion service

As part of application startup, OpenKM executes an OpenOffice or LibreOffice service. That's used internally for conversion purposes, for example to converting doc files to pdf.

OpenOffice or LibreOffice service can use a lot of hardware resources (CPU top 100%), that can decrease the performance of the application. A good practice is move OpenOffice or LibreOffice conversion service to another server.

Field / Property	Type	Description
system.openoffice.server	String	URL to OpenOffice or LibreOffice service. <code>http://192.168.1.34:8080/converter/convert</code>
system.openoffice.tasks	String	Restart service after x conversions to prevent memory leaks. 5

For more information read:

- [OpenOffice LibreOffice configuration parameters](#)
- [OpenOffice and LibreOffice configuration](#)

Antivirus checker

There are several configuration parameters to adapt the antivirus checker to your needs. Antivirus checker is one of the process that can use a lot of hardware resources (memory and CPU's). Aggressive antivirus checker policy can decrease the performance of the application and affect the end user feeling with user interface. The Antivirus checker works in live mode (as uploading process antivirus check the document and if a virus is detected a warning error is immediately raised and the document is not incorporated to the repository).



Antivirus checker in live mode can dramatically decrease the user interface and the end user feeling. Take in consideration that to analyse some documents the antivirus can check it in 3 seconds or more that will be added to the uploading time process.

Field / Property	Type	Description
system.antivir	String	Set the antivirus path.

Re-indexing the whole Lucene repository

Application can use two different ways of rebuilding Lucene indexes: sequential or parallel. By default sequential re-indexing is enabled, but you can select the mode using the "**hibernate.indexer.mass.indexer**" configuration parameter.

Field / Property	Type	Description
hibernate.indexer.mass.indexer	Boolean	When is enabled the rebuild Lucene indexes process goes into " Parallel mode ".

Mode	Description
Sequential mode	<p>You can use the "hibernate.indexer.batch.size.load.objects" configuration parameter to indicate to Hibernate how many object should handle every time. To avoid OutOfMemory problems, the repository needs to be re-indexed in batch. If the value of this property is too low, the performance will be bad but if is too high you can have OutOfMemory problems.</p> <ul style="list-style-type: none"> • hibernate.indexer.batch.size.load.objects: batch size used to load the root entities. <div>30</div>
Parallel mode	<p>You have several configuration properties to tune its performance:</p> <ul style="list-style-type: none"> • hibernate.indexer.batch.size.load.objects: batch size used to load the root entities. <div>30</div> <ul style="list-style-type: none"> • hibernate.indexer.threads.subsequent.fetching: number of threads used to load the lazy collections related to the indexed entities. <div>8</div> <ul style="list-style-type: none"> • hibernate.indexer.threads.load.objects: number of threads used to load the root entities. <div>4</div> <ul style="list-style-type: none"> • hibernate.indexer.threads.index.writer: number of threads used to analyse the documents and write to the index. <div>3</div>

Execution timeout

The Application executes external applications to process documents, for example extracts the text with an OCR engine, analyse with an Antivirus software, transforms documents to other formats among other actions. Sometimes these processes can take a lot of time or for some reason is not finished correctly and the process keep on the OS consuming resources. To prevent it, the parameter "**system.execution.timeout**" set the maximum allowed time of execution on external applications.

Field / Property	Type	Description
system.execution.timeout	Integer	Set the maximum allowed time of execution on external applications. The units are minutes. By default the configuration is set to 5 minutes. <div>5</div>

Uploading bandwidth

Sometimes you want to restrict the total bandwidth used by each user while uploading files.

Field / Property	Type	Description
upload.throttle.filter	Boolean	Limit the total uploading bandwidth to 10kb/sec per user.

Remote conversion service

On huge repository with a lot of concurrent users is a good idea to configure a server only for document transformation.



To enable this feature is needed installing a specific OpenKM service application. Contact with OpenKM technical staff for more information.

Field / Property	Type	Description
remote.conversion.server	String	Set the URL of the conversion service. <div>http://192.168.2.1:8080/converter/convert</div>

Activity log actions

OpenKM can log a lot of information related to the activity of the users , but sometimes these actions don't need to be logged and fill your activity log table.



The table where activity log is stored - OKM_ACTIVITY - may grow quickly storing millions of records.

There is a configuration property name "**activity.log.actions**" where you can set which actions to log. By default this is set to the

most common or interesting actions. You can use regular expressions to define these actions. Read [Java Regex Tutorial](#) for more info about Java regular expressions.

Field / Property	Type	Description
activity.log.actions	List	<div><i>LOLOGIN</i> <i>LOGOUT</i> <i>CREATE_.*</i> <i>DELETE_.*</i> <i>PURGE_.*</i> <i>MOVE_.*</i> <i>COPY_.*</i> <i>CHECKOUT_DOCUMENT</i> <i>CHECKIN_DOCUMENT</i> <i>GET_DOCUMENT_CONTENT.*</i></div>



For a complete list of action see [Activity log](#).

OpenOffice & LibreOffice configuration parameters

As part of the application startup, OpenKM executes an OpenOffice or LibreOffice service. That's used internally for conversion purposes, for example to converting doc files to pdf.



The OpenOffice or LibreOffice process started before OpenKM startup, can cause the failure of the service that OpenKM internally executes during startup.



The OpenOffice or LibreOffice service can use a lot of hardware resources (CPU top 100%).

Field / Property	Type	Description
system.openoffice.path	String	Path to OpenOffice or LibreOffice.
system.openoffice.dictionary	String	Dictionary used by search engine tokenizer.
system.openoffice.server	String	URL to OpenOffice or LibreOffice service. <div> This parameter replaces the use of system.openoffice.path parameter. </div> <div>http://192.168.1.34:8080/converter/convert</div>
system.openoffice.tasks	String	Restart service after x conversions to prevent memory leaks. <div>5</div>
system.openoffice.port	String	Default port is 2002. <div>2002</div>

WebDAV configuration parameters

Field / Property	Type	Description
system.webdav.server	Boolean	Enable WebDAV service. By default is disabled. <div>false</div>
system.webdav.fix	Boolean	Character ":" can cause problems with some WebDAV client with this parameter enabled replaces in all URL the character ":" to "_". By default is disabled. <div>false</div>





CSV configuration parameters




The CSV ("Comma Separated Value") file format is often used to exchange data between disparate applications. The file format, as it is used in Microsoft Excel, has become a pseudo standard throughout the industry, even among non-Microsoft platforms.





Some application features import or export data in CSV (Comma Separated Value) like results in search view users in administration among others. The format of the output CSV file can be configured with several parameters to your needs.






Field / Property	Type	Description
csv.format.delimiter	String	Set the delimiter used between columns. By default is used character ";".
csv.format.ignore.empty.lines	Boolean	Set if empty lines must be processed or not. By default the value is false .
csv.format.quote.character	String	Set the quote character used to enclose column values. By default is set to " character.
csv.format.skip_header	Boolean	Set if first row contains column names and must not be processed as values or not. By default is set to true .
csv.format.comment.indicator	String	Set character used to indicate the beginning of a comment. By default is set to #.

Other configuration parameters


Property	Type	Description
system.ocr.rotate	String	<p>This parameter forces OCR to process on the document rotated.</p> <div>  <p>The feature is useful for example in upside down scanned standard OCR direction will fails.</p> </div> <p>The parameter is a collection of degrees separated by character ";"</p> <div> <p>90;180;270;</p> </div> <p>More information at Configuring OCR engine.</p>
system.document.name.mismatch.check	Boolean	<p>This parameter is used to enable or disable the name check when a new document is uploaded.</p>
max.search.results	Integer	<p>This parameter sets the limits of the search results. For example a value will never get more than 500 results of any query.</p> <div> <p>500</p> </div> <div>  <p>The reason why the number of results is limited is that so retrieve a huge number of results, specially when the query across api methods.</p> </div>
system.demo	Boolean	<p>This parameter is used to switch openkm to demo mode.</p> <div>  <p>On production environment should never be enabled.</p> </div>
system.keyword.lowercase	Boolean	<p>This parameter is used to force all keyword to lowercase.</p> <div>  <p>The application is case sensitive, that means a user can search like "keyone" and "KeyOne" and for the application these are distinct. To prevent this problem is always good idea force keyword to lowercase.</p> </div>

tomcat.connector.uri.encoding	String	<p>By default Tomcat Connector encoding is set to "ISO-8859-1" but you can change it with this parameter.</p> <div>  <p>When you set the "URIEncoding" parameter in Tomcat connector you also need to set this parameter to "UTF-8".</p> <pre>URIEncoding="UTF-8"</pre> </div> <pre>ISO-8859-1</pre>
mobile.theme	String	<p>There're five mobile themes ("a", "b", "c", "d", "e").</p> <div> <p>More information about themes at jQuery mobile themes.</p> </div> <pre>b</pre>
plugin.debug	Boolean	When enabled show debug information about application plugin features.
principal.hide.connection.roles	Boolean	Hide "ROLE_USER" and "ROLE_ADMIN" roles from available roles list.
		<div>  <p>By default the hidden roles are set by "default.user.role" and "default.admin.role" parameters value. Usually "ROLE_USER" and "ROLE_ADMIN".</p> </div>
principal.database.filter.inactive.users	Boolean	Hide inactive users for available users list.
restrict.file.name	String	<p>Restrict file name extension that are not allowed.</p> <div>  <p>Each file name extension is separated by character ";".</p> </div> <pre>*~,*.bak</pre>
restrict.file.mime	Boolean	Restrict file name extension based on existing ones at Administration > Configuration > File Management .
native.sql.optimizations	Boolean	For best performance, some database operations are done with SQL instead of Java.



repository.stats.optimization	Boolean	When is enabled the Administration > Statistics results need less time be calculated.
repository.content.checksum	Boolean	When is enabled is calculated the document checksum for each upload.
system.pdf.force.ocr	Boolean	Force to processing all PDF files with the OCR engine.
store.node.path	Boolean	When enabled is stored in database the path of each node. <div>  For performance reason on move, delete and rename operations we suggest to do not enable this parameter. We only recommend on repositories with less 100K nodes. </div>
shCoreEclipse.css	String	CSS style used by syntax highlighter. <div>  More information at http://alexgorbatchev.com/SyntaxHighlighter </div>
shThemeEclipse.css	String	CSS style used by syntax highlighter. <div>  More information at http://alexgorbatchev.com/SyntaxHighlighter </div>
system.catdoc.xls2csv	String	Native XLS to CSV converter tool. Used in NativeMsExcelTextExtra and other Ubuntu based distros is included in the catdoc package. <div> <pre>/usr/bin/xls2csv \${fileIn}</pre> </div>
system.maintenance	Boolean	When is enabled the application goes to "maintenance mode", that means the okmAdmin user will be able to login.
system.profiling	Boolean	The Profiling stats feature gives low level value information to the administrator to detect application performance problems. <div>  You also can enable profiling from Administration > Utilities > Stats . </div>

<i>user.keywords.cache</i>	Boolean	<p>When enabled the keyword of the nodes are cached.</p> <div>  The feature provide best performance with less database c </div>
<i>user.item.cache</i>	Boolean	<p>When enabled the application cached the number of nodes owned by 1</p> <div>  The feature provide best performances with less database calculating the nodes owned by each user. </div>
<i>max.search.results</i>	Integer	<p>Set the limit of the results returned by a search to the repository.</p>
<i>html.syntaxhighlighter.core</i>	String	<p>Set the syntax highlight css core.</p> <div>  More information at SyntaxHighlighter. </div>
<i>html.syntaxhighlighter.theme</i>	String	<p>Set the syntax highlight css theme.</p> <div>  More information at SyntaxHighlighter. </div>
<i>kea.thesaurus.skos.file</i>	String	<p>The SKOS file.</p> <div> /vocabulary/ag_skos_20070219.rdf </div> <div>  When enabled restricts the number of keywords available with nodes. The SKOS file must be based on the ontology ontology file) set at kea.thesaurus.owl.file property (Th configuration). More information about SKOS (Simple Knowledge Orga). </div>
<i>kea.thesaurus.vocabulary.serql</i>	String	<p>Query to retrieve SKOS terms.</p> <div> <pre>SELECT X,UID FROM {X} skos:prefLabel {UID} WHI lang(UID) ="en" USING NAMESPACE rdf=</pre> </div>

		<pre><http://www.w3.org/1999/02/22-rdf-syntax-ns#>, <http://www.w3.org/2004/02/skos/core#>, rdfs= <http://www.w3.org/2000/01/rdf-schema#>, dc= <http://purl.org/dc/elements/1.1/>, dcterms= <http://purl.org/dc/terms/>, foaf= <http://xmlns.com/foaf/0.1/></pre>
registered.text.extractors	List	<p>List of enabled text extractor</p> <pre>org.apache.jackrabbit.extractor.PlainTextExtractor org.apache.jackrabbit.extractor.MsWordTextExtractor org.apache.jackrabbit.extractor.MsExcelTextExtractor org.apache.jackrabbit.extractor.MsPowerPointTextExtractor org.apache.jackrabbit.extractor.OpenOfficeTextExtractor org.apache.jackrabbit.extractor.RTFTextExtractor org.apache.jackrabbit.extractor.HTMLTextExtractor org.apache.jackrabbit.extractor.XMLTextExtractor org.apache.jackrabbit.extractor.MsOutlookTextExtractor com.openkm.extractor.PdfTextExtractor com.openkm.extractor.AudioTextExtractor com.openkm.extractor.ExifTextExtractor com.openkm.extractor.Tesseract3TextExtractor com.openkm.extractor.SourceCodeTextExtractor com.openkm.extractor.MsOffice2007TextExtractor</pre> <div>  Take a look at Creating your own Text extractor. </div>
version.append.download	Boolean	Append the document version in the document file name when downloading from OpenKM.
rss.news	Boolean	Enable or disable news in the login page.
rss.news.box.width	Integer	Width of the news box in the login page.
		300

<i>rss.news.max.size</i>	<i>Integer</i>	<p>Maximum number of news retrieved from the news server.</p> <div>10</div> <div>  <p>News are cached from OpenKM server and refreshed once a day. That means any change on this parameter will take effect after restarting OpenKM service.</p> </div>
<i>rss.news.visible</i>	<i>Integer</i>	<p>Number of visible news at the same time.</p> <div>1</div>
<i>webservices.visible.properties</i>	<i>List</i>	<p>List of configuration properties what are granted to view by users with grants from webservices.</p>

Experimental configuration parameters

Property	Type	Description
<i>unit.testing.folder</i>	String	Folder used by testing process. <code>/okm:root/okmTesting</code>
<i>unit.testing.user</i>	String	User used for unit testing. <code>okmAdmin</code>
<i>unit.testing.password</i>	String	Password of the user used for unit testing. <code>admin</code>
<i>unit.testing.url.base</i>	String	The unit test URL. <code>http://localhost:8080/OpenKM/</code>
<i>amazon.access.key</i>	String	The Amazon s3 access key.  Must be configured to export repository content to Amazon S3.
<i>amazon.secret.key</i>	String	The Amazon s3 secret key.  Must be configured to export repository content to Amazon S3.

AutoCAD configuration parameters

These parameters are used in AutoCAD integration.

Property	Type	Description
system.dwg2dxf	String	DWG to DXF converter tool. Used in old AutoCAD preview. <div>wine /home/openkm/.wine/ACC/AcmeCADConverter.exe</div>

Thesaurus configuration

A **thesaurus** is a book that lists **words grouped together according to similarity of meaning** (containing synonyms and sometimes antonyms), in contrast to a dictionary, which contains definitions and pronunciations.

In Information Science, Library Science, and **Information Technology**, **specialized thesauri are designed for information retrieval**. They are a type of controlled vocabulary, for indexing or tagging purposes. If you want more information about the term you can take a look at <http://en.wikipedia.org/wiki/Thesaurus>.



OpenKM can work with thesaurus defined in **.owl** and **.rdfs** formats.

There's a great utility <http://protege.stanford.edu/> for creating and maintaining your own thesaurus. Specific thesauri can be found on the Internet. For example, [Agrovoc](#) (FAO), [NASA Thesaurus](#), [UNESCO Thesaurus](#), [Public administration Thesaurus](#) among others. Some of them are free and some have usage restrictions.

In order to correctly configure OpenKM to use a thesaurus the following configuration properties must be set:

Property	Description
kea.thesaurus.owl.file	<p>The ontology file (Web ontology file)</p> <pre>vocabulary/file.owl</pre>
kea.thesaurus.base.url	<p>The base URL.</p> <pre>http://www.someweb.org</pre>
kea.thesaurus.tree.root	<p>Query to get the first node.</p> <pre>SELECT DISTINCT UID, TEXT FROM {UID} Y {OBJECT}, {UID} rdfs:label {TEXT} ; [rdfs:subClassOf {CLAZZ}] where not bound(CLAZZ) and lang(TEXT)="en" USING NAMESPACE foaf= <http://xmlns.com/foaf/0.1/>, dcterms= <http://purl.org/dc/terms/>, rdf=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>, owl= <http://www.w3.org/2002/07/owl#>, rdfs= <http://www.w3.org/2000/01/rdf-schema#>, skos=<http://www.w3.org/2004/02/skos/core#>, dc= <http://purl.org/dc/elements/1.1/></pre>

<i>kea.thesaurus.tree.children</i>	<p>Query to get children node.</p> <pre>SELECT DISTINCT UID, TEXT FROM {UID} rdfs:subClassOf {CLAZZ}, {UID} rdfs:label {TEXT} where xsd:string(CLAZZ) = "RDFparentID" and lang(TEXT)="en" USING NAMESPACE foaf= <http://xmlns.com/foaf/0.1/>, dcterms= <http://purl.org/dc/terms/>, rdf=<http://www.w3.org /1999/02/22-rdf-syntax-ns#>, owl= <http://www.w3.org/2002/07/owl#>, rdfs= <http://www.w3.org/2000/01/rdf-schema#>, skos=<http://www.w3.org/2004/02/skos/core#>, dc= <http://purl.org/dc/elements/1.1/></pre>
---	---

Take a look a full configuration example at [Creating your own thesaurus](#).

Some interesting URL:

- [A Method for Converting Thesauri to RDF/OWL](#)
- [OWL to SKOS online conversor](#)
- [SKOS to OWL online conversor](#)
- [SKOS Simple Knowledge Organization System](#)

Datastore

By default OpenKM store the binary information of the documents in the file system.



Usually you will get the best performance storing document in the file system. Basically with this configuration, when you upload or download a document it's open a file system stream directly from the server.

The parameter **repository.datastore.backend** help you to indicate what datastore will be used by OpenKM. You must change the configuration properties to be set at \$TOMCAT_HOME/OpenKM.cfg file and restart OpenKM to take effect.



You can not change datastore after you have uploaded documents into OpenKM .

You can only change it only at the first start up of the application or if you still have not uploaded documents.

File system datastore

The file system datastore is the default datastore and what we suggest for almost OpenKM installations.

```
repository.datastore.backend=fs
```

Database datastore

When database datastore is enabled, the binary information of the documents are stored into the database.



Usually with this configuration you should get less performance. Basically when you upload or download a document, it's created a temporary file in the OpenKM server what is uploaded in the database, that means more steps and extra delay for setting and retrieving binary data from database.

```
repository.datastore.backend=db
```

MIME types

The application supports several documents MIME type. Every MIME type has one or several file extensions which are used to match the uploaded document MIME type.



Based on the MIME type associated to the uploaded document, the application chooses the associated text extractor. Text extractor works only for a few MIME types that can easily be extended.

For example, if you upload a "sample.doc", it's detected MIME type is "application/msword" because its extension is ".doc". In this case, the text extractor class is "com.openkm.extractor.MsWordTextExtractor".


The detected document MIME type is also used to display an icon which represents the type of document. You can change this icon if you like.

There is list of common documents MIME types already registered in OpenKM.



If you upload a document which MIME type is not in the list, it will be detected as "application/octet-stream".

Add new MIME type

- Click **new MIME Type** icon .
- In Create mime type view **fill the required fields**:




Field	Value
Name	<p>The MIME type name</p> <input type="text" value="application/epub+zip"/>
Description	<p>General purpose description.</p> <input type="text" value="EPUB"/>
Extensions	<p>Associated document extensions.</p> <div>  In case multiple extensions, each one should be separated by character "," </div> <input type="text" value="epub"/>
	<p>Image file.</p>


Image	 <i>Image size should be 16x16 pixels.</i>
Search	<i>When is checked it means it will be shown in the application user interface at search view MIME type list.</i>

- Click **Create button**.

Edit MIME Type

- Click **Edit icon** .
- In **Edit MIME type view** make changes in fields.
- Click **Edit button**.

Delete MIME type

- Click **Delete icon** .
- In **Delete MIME type view** click **Delete button**.


Backup MIME types

- Click at top right the **Export Mime type icon** .
- After a few seconds the application will download an SQL file.

Restore MIME types



Restoring MIME types will remove actual values and will be replaced by the original ones.

- Go to **bottom page**.
- Select **MIME type sql file**.
- Click  **Import MIME types button**.

Statistics

In the statistics view, a graphical information about repository size and its distribution on the distinct root nodes (taxonomy, personal, trash, etc.) is shown. Also is shown the basic server information like memory and disk usage.

To understand how your repository is growing and making hard disk estimations, these pie charts are interesting:

- Documents by context.
- Documents size by context.
- Folders by context.

Some pie charts describe the distribution by context. These contexts are:

- Taxonomy (the /okm:root context)
- Template (the /okm:template context, where you can place your document templates)
- Personal (the /okm:personal context, where each OpenKM user can stores their personal - not shared - documents)
- Trash (the /okm:trash context, which stores the deleted document, folders, etc.)



If the trash context is too large the trash bin should be emptied. Once you are sure these documents and folders can be really deleted, you should empty your trash or individually purge the documents. Remember that each user has its own trash and every user is responsible of purging their own one.

Check the status of the Pending task queue

Pending tasks are only used in some application sections where an user action can takes longer than usual.

- Click at top right the menu option **Pending task queue**.
- In **Pending task queue view** is shown the status of the tasks. This is a list of the task to be processed or in progress.



For example, if you want to change the permissions in a folder with lots of documents and / or a very depth structure this actions will be promoted to pending task. Meanwhile the user can do more thinks while this one is executed in background. If the number of document to process is huge, this task may take hours to complete.

Check the status of the Text extraction queue

When upload a document to the application, its content is not extracted immediately but enqueued for this.

- Click at top right the menu option **Pending text extractor queue**.
- In **Pending text extractor queue view** is shown the status of the text extractor process. This is a list of the documents to be processed or in progress.

More information at [Performance configuration parameters](#).



Sometimes you might be interested on process all the documents again into the text extraction queue (you have introduced new text extractor, etc ...), you can find more information at [Database description](#) about how doing it.

Scripting

Scripting was an advanced feature that enables administrators to execute some BeanShell scripts in folders, triggered on every notified event (for example uploading documents).

This feature has been enhanced by [Automation](#) (see ExecuteScripting at [Built in Automation Action plugins](#)), which helps you to do the same things with a few mouse clicks. For compatibility reasons there is also an [Automation](#) task where you can put a BeanShell code.



OpenKM uses BeanShell. For more information check <http://www.beanshell.org/intro.html>.



BeanShell is a small, free, embeddable Java source interpreter with object scripting language features, written in Java. BeanShell dynamically executes standard Java syntax and extends it with common scripting conveniences such as loose types, commands, and method closures like those in Perl and JavaScript.

See some examples at [Scripting samples](#).

Execute script as part of application starting process

Can be executed as an script as a part of application starting process. To do so, just create a **start.bsh** file in **\$TOMCAT_HOME**.



For example, we use this feature to create a complete environment (create custom users, register property groups, register workflow) each time the OpenKM demo is restarted.

Execute script as part of application stopping process

Can be executed a an script as part of the application stopping process. To do so, just create a **stop.bsh** file in **\$TOMCAT_HOME**.

Scripting in Automation

In the Automation Scripting execution, there're several variables that are directly injected in scripting environment that can be used.

List of variables what can be used directly:

Variable	Description
systemToken	The system token.

<i>node</i>	<i>Involved node (NodeDocument, NodeFolder, etc.)</i>
<i>uuid</i>	<i>Node unique identifier.</i>
<i>file</i>	<i>Only available when converting a document to PDF or to SWF.</i>
<i>userId</i>	<i>The user id.</i>

Example:

```
import com.openkm.api.*;

// Variable uuid is injected by automation to the scripting environment
OKMProperty.getInstance().addKeyword(null, uuid, "test");
```



Metadata

In the Metadata view are shown the registered metadata groups. In this view, it can create, modify or remove metadata definitions.

Property groups are shown in a table structure. First row (three columns) corresponds to the general Property group definition. The next rows are fields definition in Property group.

Group label		Group name		Group info	
Label	Name	Width	Height	Element	Other


Description:

Property	Description
Group label	Group label is shown in the user interface at Property group tab.
Group name	Unique group name identifier. <div>  Two property groups cannot have the same name. </div>
Group info	Shows information about: <ul style="list-style-type: none"> • Visibility (property group is hidden from user interface). • Read only (property group can not be modified from user interface).
Field label	Field label is shown in the user interface in the Property group tab,
Field name	Unique property group field identifier. <div>  Two property group fields can't have the same name. </div>
Field width	This parameter is used for rendering the width of the field in the user interface view.

Field height	<i>This parameter is used for rendering the height of the field in the user interface view.</i>
Element	<p><i>This parameter sets the element type. It's used for rendering several values in the field in the user interface view.</i></p> <p><i>Values:</i></p> <ul style="list-style-type: none"> • <i>CheckBox</i> • <i>Input</i> • <i>Iframe</i> • <i>Select</i> • <i>Separator</i> • <i>SuggestBox</i> • <i>Text</i> • <i>TextArea</i>
Others	<i>Shows field extra information.</i>

Edit metadata

To add, modify or delete metadata there's only one operation for all, edit the metadata xml definition.

- Click top right  **Edit metadata**.
- In Edit view, **modify the XML**.
- At bottom right, click **Edit button**.
- At top right click, click **Register metadata groups** to apply the changes.

Metadata XML definition

The application metadata is based on a formal XML definition. The DTD (Document Type Definition) defines the structure and the legal elements and attributes of an XML document.



You can see latest DTD in our website <http://www.openkm.com/dtd/property-groups-{major}.{minor}.dtd> where {major} and {minor} values are two numbers (major release and minor release). For example the location for version 2.3 of the DTD is <http://www.openkm.com/dtd/property-groups-2.3.dtd>



Each application version supports only a couple of DTD definitions, read [Application version compatibility table](#) to see which are supported by your application.

Basic XML skeleton

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.3//EN"
    "http://www.openkm.com/dtd/property-groups-2.3.dtd">
<property-groups>
</property-groups>
```




- The **DOCTYPE** is a formal definition of some DTD. In the previous example the **property-groups-2.3.dtd**.
- The XML begins with **<property-groups>** tag and ends with **</property-groups>**. Into this hierarchy are defined the metadata groups.

Metadata group definition

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.3//EN"
    "http://www.openkm.com/dtd/property-groups-2.3.dtd">
<property-groups>
  <property-group label="Consulting" name="okg:consulting">
  </property-group>
</property-groups>
```

- Each **<property-group>...</property-group>** block identify the **beginning and end** of new **metadata group definition**.
- Each metadata group has a **label** - used by user interface - **Consulting**.
- Each metadata group has a **unique name**, **okg:consulting**.
- The metadata group name always must starts with **"okg:"** (**OpenKM Group**).
- Must not use special characters for name.

Attribute	Description	Required
		true

label	The text shown as a label in the user interface.	
name	<p>Unique field identifier.</p> <div>  <p>Two metadata groups can't have the same name. Name must be unique.</p> <p>The name must start with "okg:".</p> <p>Please, use only letters, numbers and underscore: "0-9a-zA-Z_".</p> </div>	true
visible	<p>Show or hide the metadata.</p> <p>Allowed values are:</p> <ul style="list-style-type: none"> • true. • false. <div>  <p>By default the metadata are visible.</p> </div>	false
readonly	<p>Metadata values can not be modified from user interface only from API.</p> <p>Allowed values are:</p> <ul style="list-style-type: none"> • true. • false. <div>  <p>By default the metadata are modifiable.</p> </div>	false

Basic example

Available field types:

- Checkbox
- Input
- Iframe
- Select
- Separator
- Suggestbox
- Text

- `TextArea`

More information about field type elements at [Metadata fields](#)



It's a good practice to use the metadata group name as the basis to create metadata field name. For example **okp:consulting.code** use **okg:consulting** as base (changing **okg** to **okp**). That helps to find out which is the metadata group associated to each metadata field.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.3//EN"
    "http://www.openkm.com/dtd/property-groups-2.3.dtd">

<property-groups>
  <property-group label="Consulting" name="okg:consulting">
    <input label="Input label" name="okp:consulting.input1" />
    <separator label="Separator label" name="okp:consulting.separator" />
    <input label="Input label" name="okp:consulting.input2" />
  </property-group>
</property-groups>
```

Using file system DTD

If the server has no access to the Internet, then the DTD file must be accessible in some way, for example can store a copy in your application server.



The XML definition must be changed to refer to the new DTD location, otherwise you will get an application error while processing DTD file.

Can be a file system (example below) or a valid intranet URL.

```
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.3//EN"
    "file:///home/openkm/property-groups-2.3.dtd">
```

- In the example, **/home/openkm/property-groups-2.3.dtd** is the file system path to the file.

Metadata fields

These are the available field types:

Field type	Description	Visible in the Metadata Stack Panel
Checkbox	Metadata Checkbox field	Yes.
Input	Metadata Input field	No.
Select	Metadata Select field	Yes.
Separator	Metadata Separator field	No.
Suggestbox	Metadata Suggestbox field	Yes.
Text	Metadata Text field	No.
TextArea	Metadata TextArea field	No.

Some field types can have children elements, validator or option elements. This hierarchy is permitted:


- metadata group
 - Checkbox
 - Validator
 - Input
 - Validator
 - Iframe
 - Select
 - Option
 - Validator
 - Separator
 - Suggestbox
 - Validator
 - Text

- *TextArea*
 - *Validator*

Metadata Checkbox field

Checkbox



Best used to represent boolean data:


Attribute	Description	Required
label	The text shown as a label in the user interface.	true
name	<p>Unique field identifier</p> <div>  <p>Two metadata can't have the same name. Name must be unique.</p> <p>The name must start with "okp:".</p> <p>It's a good practice to use a metadata group name as basis for a metadata field name. For example use "okg:consulting" has been used for field name "okp:consulting.checkbox", changing "okg" to "okp". That helps to find out the metadata group based on metadata field name.</p> <p>Please, use only letters, numbers and underscore: "0-9a-zA-Z_".</p> </div>	true
width	The width of the HTML element.	false
height	The height of the HTML element.	false
description	General description.	false
visible	Set field will be shown or not in the user interface. By default the value is " true ".	false
readonly	Set field will not be able to be modified in user interface. By default the value is " false ".	false
Child elements: <ul style="list-style-type: none"> Validator. For more information see Metadata Validator element. 		

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.3//EN"
    "http://www.openkm.com/dtd/property-groups-2.3.dtd">
<property-groups>
```

```
<property-group label="Consulting" name="okg:consulting">  
  <checkbox label="Checkbox label" name="okp:consulting.checkbox" />  
</property-group>  
</property-groups>
```

Metadata Select field

Attribute	Description	Required
label	The text shown as label in the user interface.	true
name	<p>Unique field identifier.</p> <div>  <p>Two metadatas can't have the same name. Name must be unique.</p> <p>The name must start with "okp:".</p> <p>It's a good practice to use metadata group name as basis for metadata field name. For example use "okg:consulting" has been used for field name "okp:consulting.select", changin "okg" to "okp". That helps to find out the metada group based on metadata field name.</p> <p>Please, use only letters, numbers and underscore: "0-9a-zA-Z_".</p> </div>	true
type	<p>Type value. Allowed types:</p> <ul style="list-style-type: none"> • simple (single choice allowed). • multiple (multiple choices allowed). 	true
table	<p>Metadata table name.</p> <div>  <p>The application can store values into a metadata table, that can be used in this query.</p> </div>	*
width	The width of the HTML element.	
height	The height of the HTML element.	
optionsQuery	A metadata query used to get the selected options.	*
className	A class used to get the selected options.	*
	When adding metadata to documents, the application can suggest automatically some values based on document content analysis. By default the application has some	

suggestion	<p>implementations that can easily be extended.</p> <ul style="list-style-type: none"> • com.openkm.form.suggestion.DocumentContentContainsSuggestion: Gets suggestions based on document's content (use <code>String.contains</code>) • com.openkm.form.suggestion.DocumentContentTokenizerSuggestion: Gets suggestions based on document's content (Use <code>StringTokenizer</code>). • com.openkm.form.suggestion.DocumentTermsSuggestion: Gets suggestions based on document's term vectors. <p>For more information read Creating your own Suggestion Analyzer.</p>	false
visible	Set field will be shown or not in the user interface. By default the value is " true ".	false
readonly	Set field will not be able to be modified in the user interface. By default the value is " false ".	false
<p>Child elements:</p> <ul style="list-style-type: none"> • Option. Couples of XML <code><option label="label text" value="unique identifier value"/></code>. • Validator. For more information see Metadata Validator element. 		
<p>* There're two options to get option values:</p> <ul style="list-style-type: none"> • Directly defined in XML. • Metadata table (table + optionsQuery). <div style="border: 1px dashed orange; padding: 10px; margin-top: 10px;"> <p> One of these options must be defined, otherwise you will get an error. It can not done as a combination of several of them, only one can be choosen.</p> </div>		

Basic select example

- Type is equal to **simple**.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.3//EN"
    "http://www.openkm.com/dtd/property-groups-2.3.dtd">
<property-groups>
  <property-group label="Consulting" name="okg:consulting">
    <select label="select label" name="okp:consulting.select" type="simple">
      <option label="one" value="001" />
      <option label="two" value="002" />
      <option label="three" value="003" />
    </select>
  </property-group>
</property-groups>
```

```
</property-groups>
```

Allow multiple choices

- Type is equal to **multiple**.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.5//EN"
    "http://www.openkm.com/dtd/property-groups-2.5.dtd">
<property-groups>
  <property-group label="Consulting" name="okg:consulting">
    <select label="select label" name="okp:consulting.select" type="multiple">
      <option label="one" value="001" />
      <option label="two" value="002" />
      <option label="three" value="003" />
    </select>
  </property-group>
</property-groups>
```

Options based on a metadata table

- The application uses two tables named `OKM_DB_METADATA_TYPE` AND `OKM_DB_METADATA_VALUE` that helps storing the options values used in select. That's very useful when you have a large number of values.
- In the table `OKM_DB_METADATA_TYPE` are stored metadata table definition (table name, column name and column type).
- In the table `OKM_DB_METADATA_VALUE` are stored record values.

Create metadata table definition.



The **table** will be called **country**, with two columns, **country_id** and **country_name**.

The **country_id** column is an **integer** type and **country_name** is **text** type.

The **country_id** is set as **column 0** with **col00** value.

The **country_name** is set as **column 1** with **col01** value.

- Go to **Administration > Database query**.
- At bottom right select **JDBC** from list.
- **Paste the SQL** in the box

```
DELETE FROM OKM_DB_METADATA_TYPE WHERE DMT_TABLE='country';
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
VALUES ('country', 0, 'integer', 0);
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
VALUES ('country', 1, 'text', 0);
```

- Click **Execute button** at the bottom right the .

Insert metadata table values:

- Go to **Administration > Database query**.
- At the bottom right select **JDBC** from list.
- **Paste the SQL** in the box

```
DELETE FROM OKM_DB_METADATA_VALUE WHERE DMV_TABLE='country';
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('country',
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('country',
```

- Click the **Execute button** at the bottom right .



The insert queries depending on your database can not be a match, for example in Oracle:

```
INSERT INTO OKM_DB_METADATA_TYPE (DMT_ID, DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE,
DMT_VIRTUAL_COLUMN) VALUES (HIBERNATE_SEQUENCE.NEXTVAL, 'country', 'col00', 'integer',
'country_id');
```

More information at [Creating your own database tables](#).

Finally the XML definition:

- In optionsQuery the values that begin with the character \$ are identified as column names. See **\$country_id** and **\$country_name** below.
- Can use normal SQL clauses like "**order by**", etc.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.5//EN"
"http://www.openkm.com/dtd/property-groups-2.5.dtd">
<property-groups>
  <property-group label="Consulting" name="okg:consulting">
    <select label="country" name="okp:consulting.country" type="simple" table="count
      optionsQuery="select $country_id, $country_name from DatabaseMetadataVal
    </property-group>
  </property-groups>
```

Suggested options

Based on the content analyzer the application can suggest some options values. At the user interface the suggested options will be shown in green.

The application has some implementations that can easily be extended.


- **com.openkm.form.suggestion.DocumentContentContainsSuggestion:** Get suggestions based on document content (use String.contains)
- **com.openkm.form.suggestion.DocumentContentTokenizerSuggestion:** Get suggestions based on document content (Use StringTokenizer).
- **com.openkm.form.suggestion.DocumentTermsSuggestion:** Get suggestions based on document term vectors.

For more information read [Creating your own Suggestion Analyzer](#).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.3//EN"
    "http://www.openkm.com/dtd/property-groups-2.3.dtd">
<property-groups>
  <property-group label="Consulting" name="okg:consulting">
    <select label="select label" name="okp:consulting.select" type="multiple"
      suggestion="com.openkm.form.suggestion.DocumentContentContainsSuggestion">
      <option label="one" value="001" />
      <option label="two" value="002" />
      <option label="three" value="003" />
    </select>
  </property-group>
</property-groups>
```

Metadata Separator field


Draw a separator between fields.

Attribute	Description	Required
label	The text shown as a label in the user interface.	true
name	<p>Unique field identifier</p> <div>  <p>Two metadatas can't have the same name. Name must be unique.</p> <p>The name must start with "okp:".</p> <p>It's a good practice to use a metadata group name as basis for metadata field name. For example use "okg:consulting" has been used for field name "okp:consulting.separator", changin "okg" to "okp". That helps to find out the metada group based on metadata field name.</p> <p>Please, use only letters, numbers and underscore: "0-9a-zA-Z_".</p> </div>	true
width	The width of the HTML element.	false
visible	The set field will be shown or not in user the interface. By default the value is " true ".	false
readonly	The set field will not be able to be modified in the user interface. By default the value is " false ".	false

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.3//EN"
    "http://www.openkm.com/dtd/property-groups-2.3.dtd">
<property-groups>
  <property-group label="Consulting" name="okg:consulting">
    <input label="Input label" name="okp:consulting.input1" />
    <separator label="Separator label" name="okp:consulting.separator" />
    <input label="Input label" name="okp:consulting.input2" />
  </property-group>
</property-groups>
```


Metadata Suggestbox field

This is an implementation of the typical AJAX suggest box component:

Attribute	Description	Required
label	The text shown as a label in the user interface.	true
name	<p>Unique field identifier.</p> <div>  <p>Two metadatas can't have the same name. Name must be unique.</p> <p>The name must start with "okp:".</p> <p>It's a good practice to use a metadata group name as basis for metadata field name. For example use "okg:consulting" has been used for field name "okp:consulting.suggestbox", changin "okg" to "okp". That helps to find out the metada group based on metadata field name.</p> <p>Please, use only letters, numbers and underscore: "0-9a-zA-Z_".</p> </div>	true
table	<p>Metadata table name.</p> <div> <p>The application can store values into a metadata table, that can be used in this query.</p> </div>	true
width	The width of the HTML element.	false
height	The height of the HTML element.	false
filterQuery	A hibernate query to filter the data. Use {0} to place the user input.	true
valueQuery	A Hibernate query to obtain the value of a identifier. Use {0} to place the identifier.	true
filterMinLen	The minimum chars in input to begin to filter results.	true
visible	The set field will be shown or not in the user interface. By default the value is " true ".	false

readonly	The set field will not be able to be modified in the user interface. By default the value is "false".	false
<p>Child elements:</p> <ul style="list-style-type: none"> • Validator. For more information see Metadata Validator element. 		

Suggest box values are defined in metadata tables.

- Application use two tables named `OKM_DB_METADATA_TYPE` AND `OKM_DB_METADATA_VALUE` that helps storing options values used in selects. That's very useful when you have a large number of values.
- In table `OKM_DB_METADATA_TYPE` are stored metadata table definition (table name, column name and column type).
- In table `OKM_DB_METADATA_VALUE` are stored record values.

Create metadata table definition.



The table will be called **country**, with two columns, **country_id** and **country_name**.

The **country_id** column is **integer** type and **country_name** is **text** type.

The **country_id** is set as **column 0** with **col00** value.

The **country_name** is set as **column 1** with **col01** value.

- Go to **Administration > Database query**.
- At the bottom right select **JDBC** from list.
- **Paste the SQL** in the box

```
DELETE FROM OKM_DB_METADATA_TYPE WHERE DMT_TABLE='country';
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
VALUES ('country', 0, 'integer', 0);
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
VALUES ('country', 1, 'text', 0);
```

- Click at bottom right the **Execute button**.

Insert metadata table values:

- Go to **Administration > Database query**.
- At bottom right select **JDBC** from list.
- **Paste the SQL** in the box

```
DELETE FROM OKM_DB_METADATA_VALUE WHERE DMV_TABLE='country';
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('country', 0, 'country');
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('country', 1, 'country');
```

- Click at bottom right the **Execute button**.



The insert queries depending your database can not be match, for example in Oracle:

```
INSERT INTO OKM_DB_METADATA_TYPE (DMT_ID, DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE,
DMT_VIRTUAL_COLUMN) VALUES (HIBERNATE_SEQUENCE.NEXTVAL, 'country', 'col00', 'integer',
'country_id');
```


Finally the XML definition:

- In the optionsQuery the values that begins with character \$ are identified as column names. See **\$country_id** and **\$country_name** below.
- Can use normal SQL clauses like "**lower**", "**like**", "**order by**", etc.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.3//EN"
"http://www.openkm.com/dtd/property-groups-2.3.dtd">
<property-groups>
  <property-group label="Consulting" name="okg:consulting">
    <suggestbox label="country" name="okp:consulting.suggestbox" width="200px"
      table="country" dialogTitle="Choose country" filterMinLen="3"
      filterQuery="select $country_id, $country_name from DatabaseMetadataValue dmv where"
      valueQuery="select $country_id, $country_name from DatabaseMetadataValue dmv where"
    </property-group>
  </property-groups>
```

Metadata Text field


Prints a text that can contain HTML tags.

Attribute	Description	Required
label	The text shown as a label in user interface.	true
name	<p>Unique field identifier</p> <div>  <p>Two metadatas can't have the same name. Name must be unique.</p> <p>The name must start with "okp:".</p> <p>It's a good practice to use metadata group name as basis for metadata field name. For example use "okg:consulting" has been used for field name "okp:consulting.text", changin "okg" to "okp". That helps to find out the metadata group based on metadata field name.</p> <p>Please, use only letters, numbers and underscore: "0-9a-zA-Z_".</p> </div>	true
width	The width of the HTML element.	false
height	The height of the HTML element.	false
visible	The set field will be shown or not in the user interface. By default the value is " true ".	false
readonly	The set field will not be able to be modified in the user interface. By default the value is " false ".	false

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.3//EN"
"http://www.openkm.com/dtd/property-groups-2.3.dtd">
<property-groups>
  <property-group label="Consulting" name="okg:consulting">
    <text label="text label" name="okp:consulting.text" width="200px" />
  </property-group>
</property-groups>
```

Metadata TextArea field

Prints a text that can contain HTML tags.

Attribute	Description	Required
label	The text shown as a label in the user interface.	true
name	<p>Unique field identifier</p> <div>  <p>Two metadatas can't have the same name. Name must be unique.</p> <p>The name must start with "okp:".</p> <p>It's a good practice to use metadata group name as basis for metadata field name. For example use "okg:consulting" has been used for field name "okp:consulting.textarea", changing "okg" to "okp". That helps to find out the metadata group based on metadata field name.</p> <p>Please, use only letters, numbers and underscore: "0-9a-zA-Z_".</p> </div>	true
value	Default value for the component.	false
width	The width of the HTML element.	false
height	The height of the HTML element.	false
visible	The set field will be shown or not in the user interface. By default the value is " true ".	false
readonly	The set field will not be able to be modified in the user interface. By default the value is " false ".	false
<p>Child elements:</p> <ul style="list-style-type: none"> Validator. For more information see Metadata Validator element. 		

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.3//EN"
    "http://www.openkm.com/dtd/property-groups-2.3.dtd">
<property-groups>
  <property-group label="Consulting" name="okg:consulting">
```

```
<textarea label="textarea label" name="okp:consulting.textarea" width="200px" />
</property-group>
</property-groups>
```

Metadata Validator element

The goal of the Validation element is to ensure that the user provides necessary and properly formatted information needed to successfully complete an operation.

The validator can only be used in combination with other fields:

- [Metadata Checkbox field](#)
- [Metadata Input field](#)
- [Metadata Select field](#)
- [Metadata Suggestbox field](#)
- [Metadata TextArea field](#)

Available validators:

<i>name</i>	<i>Description</i>
req	Required value.
alpha	Alphanumeric value.
dec	Decimal value.
num	Numeric value.
email	E-mail value.
url	URL value.
maxlen	Set the maximum length allowed.
minlen	Set the minimum length required.
lt	Less than.
gt	Greater than.

<i>min</i>	<i>Minimum value.</i>
<i>max</i>	<i>Maximum value.</i>
<i>regex</i>	<i>Regular expression.</i>
Parent: <ul style="list-style-type: none"> • Metadata Checkbox field • Metadata Input field • Metadata Select field • Metadata Suggestbox field • Metadata TextArea field 	

Basic example

- Value required "**req**".
- Value must be decimal "**dec**".
- The maximum lenght of the decimal value is 6 digits "**maxlen**".

```
<input label="Decimal value" name="okp:consulting.decimal">
  <validator type="dec" />
  <validator type="req" />
  <validator type="maxlen" parameter="6" />
</input>
```


Regular expression

If you want a validation which is not included in these prefined, you can use a regular expression. For example, imagine you want to restrict an input to lowercase ASCII values:

```
<input label="Lowercase value" name="regExp">
  <validator type="regex" parameter="[a-z]+" />
</input>
```


Metadata Option element

The option element is only used in combination with [Metadata Select field](#). The option value represents a list of distinct values to select.

Attribute	Description	Required
label	The text shown as label in the user interface.	true
value	Unique value identifier. <div>  Two options can't have the same value. Value must be unique. </div>	true
selected	When present, it specifies that an option should be pre-selected when the form is redere.	false
Parent : <ul style="list-style-type: none"> Select. For more information, Please see: Metadata Select field. 		



Basic select with selected option

- attribute selected=true, when present, it specifies that an option should be pre-selected when the page loads.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.3//EN"
    "http://www.openkm.com/dtd/property-groups-2.3.dtd">
<property-groups>
  <property-group label="Consulting" name="okg:consulting">
    <select label="select label" name="okp:consulting.select" type="simple">
      <option label="one" value="001" />
      <option label="two" value="002" selected="true" />
      <option label="three" value="003" />
    </select>
  </property-group>
</property-groups>
```

Metadata Input field

In most cases this field is used to enter free form text, although it can be used to save http links, date, data or refer to application folder (as internal link to this folder).

Attribute	Description	Required
label	The text shown as a label in user interface.	true
name	<p>Unique field identifier.</p> <div>  <p>Two metadata can't have the same name. Name must be unique.</p> <p>The name must start with "okp:".</p> <p>It's a good practice to use metadata group name as basis for metadata field name. For example use "okg:consulting" has been used for field name "okp:consulting.input", changin "okg" to "okp". That helps to find out the metada group based on metadata field name.</p> <p>Please, use only letters, numbers and underscore: "0-9a-zA-Z_".</p> </div>	true
type	<p>Type value. Allowed types:</p> <ul style="list-style-type: none"> • text. • date. • link. • folder. 	true
value	<p>Specifies a default value for this component.</p> <div>  <p>There's a special case for date types. Data is stored in format "yyyyMMddHHmmss", be aware of this format when using OpenKM API.</p> </div>	false
width	The width of the HTML element.	false
height	The height of the HTML element.	false
visible	The set field will be shown or not in the user interface. By default the value is " true ".	false

readonly	The set field will not be able to be modified in the user interface. By default the value is " false ".	false
-----------------	--	-------

Child elements:

- Validator. For more information please see [Metadata Validator element](#).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.3//EN"
    "http://www.openkm.com/dtd/property-groups-2.3.dtd">
<property-groups>
  <property-group label="Consulting" name="okg:consulting">
    <input label="Input label" name="okp:consulting.input" />
  </property-group>
</property-groups>
```

Manage users and roles

In the Manage users and roles view you can do:

- [Manage users.](#)
- [Manage roles.](#)
- [Manage user mail accounts.](#)
- [Logged users.](#)
- [Message queue.](#)



By default the application manages users and roles in its own database.

It is also possible to connect the application to an external directory (LDAP directory for instance).

Also the login module can be extended and create your own login adapter. More information about it [Creating your own login module.](#)




When the application is connected to LDAP the manage users and roles view is automatically set to readonly mode.

In this scenario the LDAP takes full control of users and roles.

The restricted actions are:

- Create, delete or modify a user.
- Create or delete a role.

Manage users

To manage users on the top right click on  **User list button**.



By default the application manages users and roles in its own database.

It is also possible to connect the application to an external directory (LDAP directory for instance).

The login module can be extended and create your own login adapter. More information about it [Creating your own login module](#).












When the application is connected to LDAP the manage users view is automatically set to readonly mode. Almost all actions are restricted.

Users are shown in a table structure.


Filtering options							
<i>ID</i>	<i>Name</i>	<i>Mail</i>	<i>Roles</i>	<i>Profile</i>	<i>Active</i>	<i>Chat</i>	<i>Actions</i>

User parameters:


<i>Field</i>	<i>Description</i>	<i>Required</i>
<i>ID</i>	<p>The id the user will use to log in.</p> <div>  Two users can't have the same id. </div>	<i>true</i>
<i>Name</i>	User name.	<i>true</i>
<i>Mail</i>	User email address.	<i>true</i>
<i>Roles</i>	<p>Roles - groups - from which the user is a member.</p> <div>  User must be at least in 'ROLE_ADMIN' or 'ROLE_USER' role. </div>	<i>true</i>
<i>Profile</i>	Each profile is a set of options available on the user interface.	<i>true</i>

Active	Sets user enabled or not.	<i>true</i>
Chat	<p>Indicates if user is connected to the chat.</p> <div style="border: 1px dashed orange; padding: 10px; background-color: #fff9c4;">  <p>Although a user can have the application opened on several browser tabs. Only one of them will be logged on chat.</p> </div>	<i>false</i>
Action	<p>Available actions:</p> <ul style="list-style-type: none"> •  Edits a user. •  Deletes a user. •  Sets the profile. •  Configures mail accounts. More information here Manage user mail accounts. •  Configures twitter account. •  Shows the activity log of the user. 	*
* Mail account and twitter account are optional.		

Create a user

- On bottom right, click on the  **New user icon**.
- **Fill all the fields**. Remember all users must be at least in 'ROLE_ADMIN' or 'ROLE_USER' roles.
- Click on the **Create button**.

Edit a user

- Click on the  **Edit user icon**.
- **Modify the fields**.
- Click on the **Edit button**.

Delete a user

- Click on the  **Delete user icon**.
- Click on the **Delete button**.

Change the user profile


- Click on the  **User config icon**.
- Choose the profile.

- Click on the **Edit button**.




By default the profile named "default" is automatically assigned to any new user.

Export the users

- On the top right click on the  **export users**.
- The application will download a .csv file.

Enable twitter account

- Click on the  **Configure twitter account**.
- Set **twitter user**.
- **Enable active** checkbox.
- Click on the **Create button**.

Manage user mail accounts

In the user accounts view, are shown the mail user accounts. These mail accounts are used by application to periodically import mail to application repository.

To manage user mail account, from Users view click on  **Email account icon**.

Mail account hierarchy

- Mail account
 - Filter
 - Rules



- The mail account sets the data connection to the mail server to retrieve mails.
- Filters are optional and sets the application location - folder - where to store mails.
- Rules are optional and sets a couple of rules that controls what are the valid mails to import.



When a filter is not set, by default the application will store mail into /okm:mail/userid folder.







When rules are not set, by default any incoming mail is accepted.

Mail accounts are shown in a table structure.


Mail protocol	Mail host	Mail user	Mail folder	Active
----------------------	------------------	------------------	--------------------	---------------

Mail account parameters:

Field	Description	Required
Mail protocol	<p>The protocol used to connect to the mail server.</p> <p>Available options:</p> <ul style="list-style-type: none"> • IMAP • IMAPS • POP3 • POP3S 	<i>true</i>
Mail host	The mail server host.	<i>true</i>

Mail user	The user name to access the mail account.	true
Mail password	The password to access the mail account.	true
Mail folder	<p>The IMAP folder.</p> <div>  <p>Only applicable when mail protocol value is IMAP or IMAPS. In case of POP3 you can leave empty otherwise need to be set to INBOX.</p> </div>	false
Mail mark seen	<p>Sets to mail server the seen status.</p> <p>Available options:</p> <ul style="list-style-type: none"> • Keep (keep as is). • Yes (mark as seen) • No (mark as not seen). 	true
Mail mark deleted	<p>Set mail to be deleted.</p> <div>  <p>When checked the mail will be marked to be deleted from the mail server.</p> </div>	true
Active	Sets mail account enabled or not.	true
Actions	<p>Available actions:</p> <ul style="list-style-type: none"> •  Edits a mail account. •  Deletes a mail account. •  Sets mail account configuration filters. •  Browsers emails in the mail server. 	

Create a new mail account

- At the top right click on the  **Create new account icon**.
- **Fill in all the fields.**

- Click on the **Create button**.



It is a good practice before clicking on Create button, to **check the mail configuration**. There's a button named **Check** that will check the mail configuration. If all is right application will get **OK** status.

Edit a mail account

- Click on the **Edit mail account icon**.
- **Modify the fields**.
- Click on the **Edit button**.

Delete a mail account

- Click on the **Delete mail account icon**.
- Click on the **Delete button**.

Browser emails in the mail server

- Click on the **Server list icon**.



In this view mail can be directly imported.

Manage mail account filters

In the mail account configuration, rules can be set to automatically store mails on different application folders.







For example, filters can help to automatically catalog mails, based on a "subject" value or "from" value among others.

Mail accounts filter are shown in a table structure.


Folder	Grouping	Active
--------	----------	--------

Mail account filter parameters:

Field	Description	Required
Folder	Sets the destination folder.	true.
	Sets automatic catalog by year/month/day enabled.	

Grouping	 <p>To prevent having too many mails in the same folder, we recommend enabling mail grouping.</p> <p>The folders year/month/years are automatically created by application.</p>	<i>false.</i>
Active	Sets filter enabled or not.	<i>true.</i>
Actions	<p>Available actions:</p> <ul style="list-style-type: none">  Edit a filter.  Delete a filter.  Show rules. 	

At least a filter sets a destination folder it also can have additional rules.



A rule can only be set into a filter.

Mail accounts rules are shown in a table structure.



Field	Operation	Value	Active
-------	-----------	-------	--------

Mail account rule parameters:



Field	Description	Required
Field	<p>Sets the mail field that will be evaluated by the rule.</p> <p>Available options are:</p> <ul style="list-style-type: none"> From To Subject Content 	<i>true.</i>
Operation	<p>Set the operation that will be done for the evaluation process.</p> <p>Available options are:</p> <ul style="list-style-type: none"> Contains 	<i>true.</i>

	<ul style="list-style-type: none"> • <i>Equal</i> 	
Value	Sets the value that will be used for evaluation process.	<i>true.</i>
Active	Sets rule enabled or not.	<i>true.</i>



Create a mail account filter

- Click on the  **Show filters icon.**
- On top right click on  **New filter icon.**
- **Fill in all the fields.**
- Click on the **Create button.**



Edit a mail account filter

- Click on the  **Show filters icon.**
- On top right click on  **Edit filter icon.**
- **Modify the fields fields.**
- Click on the **Edit button.**


Delete a mail account filter

- Click on the  **Show filters icon.**
- On top right click on  **Delete filter icon.**
- Click on the **Delete button.**

Create a mail account rule




- Click on the  **Show filters icon.**
- Click on the  **Show rules icon.**
- On top right click on  **New rule icon.**
- **Fill in the fields.**
- Click on the **Create button.**

Edit a mail account rule

- Click on the  **Show filters icon.**
- Click on the  **Show rules icon.**
- On top right click on  **Edit rule icon.**

- **Modify the fields.**
- Click on the **Edit button**.

Delete a mail account rule

- Click on the  **Show filters icon**.
- Click on the  **Show rules icon**.
- On top right click on  **Delete rule icon**.
- Click on the **Delete button**.

Manage roles

To **manage roles** at the top right click on  **Role list button**.



By default the application manages users and roles in its own database.

It is also possible to connect the application to an external directory (LDAP directory for instance).

Also the login module can be extended and create your own login adapter. More information about it [Creating your own login module](#).






When the application is connected to LDAP the manage roles view is automatically set to readonly mode. Almost all actions are disabled.


Roles are shown in a table structure.

Id	Active	Actions
----	--------	---------


Role parameters:

Field	Description	Required
Id	<p>The id of the role.</p> <div>  Two roles can't have the same id. </div>	true
Active	Sets if role is enabled or not.	true
Action	<p>Available actions:</p> <ul style="list-style-type: none">  Edits a role.  Deletes a role. 	

Create a role

- At the bottom right, click on the  **New role icon**.
- Fill all the fields.**
- Click on the **Create button**.

Edit a role

- Click on the  **Edit role icon**.
- **Modify the fields**.
- Click on the **Edit button**.



ROLE id can't be modified to prevent repository security inconsistencies. This operation can be done only at database level.

Delete a role

- Click on the  **Delete role icon**.
- Click on the **Delete button**.



Logged users

To manage logged users on top right click on  **Logged users list button**.

Logged users are shown in a table structure :

User	Session Id	Remote IP	Remote Host	Creation	Last access
------	------------	-----------	-------------	----------	-------------

Real time user activity log

- Click on on  **Activity log icon**.
- Will be shown as a table with all actions performed by the user today.
- Click on the  **Filter button** to refresh values.

Message queue

Message queue helps administrators to send messages across the user interface.



Helps administrators to send a message to all logged users to indicate the system will be halt and automatically logout the users 5 minutes after they received the message.

The application sends messages to a user when a background process has been completed.


The application sends warnings to a user about an action that is not allowed.

Among others messages.


Queue of messages shown in a table structure:

Date	Action	Message	To All	To Users	To Roles
------	--------	---------	--------	----------	----------


Message parameters:

Field	Description	Required
Date	<p>Date of the message.</p> <div>  Automatically set by the application. </div>	true.
Action	<p>Available values:</p> <ul style="list-style-type: none"> Logout (force users who receives it to logout) 	false.
Message	The text of the message.	true.
To All	When present, the message is sent to all the users.	*
To Users	When present, indicate a set of users to sent the message.	*
To Roles	When present, indicate a set of users which are members of some of these roles.	*
* At least one of the options.		


Send a message

- Click on the  **New message icon**.
- **Fill the fields**.
- Click on the **Create button**.


Edit a message

- Click on the  **Edit message icon**.
- **Modify the fields**.
- Click on the **Edit button**.

Delete a message

- Click on the  **Delete message icon**.
- Click on the **Delete button**.


Delete all messages queue

- On top right click on  **Clear button**.



This operation can not be reverted. All the message queue will be deleted.

Refresh message queue

- On the top right click on  **Refresh button**.

Profiles

Profiles are a set of user interface configuration parameters. They help the administrators to set the available interface features for each user.



For example:

- Restricts the user interface hiding search and dashboard tabs and almost toolbar buttons.
- Restricts the user interface showing only search view.



The application and specially the user interface comes by default with a several features and extensions.

Not all users need access to everything, specially at the beginning when a lot of features can cause confusion to end user.

Our suggestion is going from less to more features as a natural process of the application knowledge.


Profiles are shown in a table structure.

Name	Active	Actions
------	--------	---------


Profiles are divided in four tabs:

- [Profile General tab](#)


Create a new profile

- Click on the  **New profile icon**.
- **Fill in the fields.**
- Click on the **Create button**.


Edit a profile

- Click on the  **Edit profile icon**.
- **Modify the fields.**
- Click on the **Edit button**.

Delete a profile


- Click on the  **Delete profile icon**.
- Click on the **Delete button**.

Clone a profile





- Click on the  ***Clone profile icon***.
- ***Modify the fields.***
- Click on the ***Edit button***.



Profile General tab

General section

Field	Description
Name	<p>The name of the profile.</p> <div>  Two profiles cannot share the same name. </div>
Active	Sets profile enabled or not.

Misc section

Field	Description
Quota limit (MB)	<p>Sets the total maximum size of documents that a user can upload.</p> <div>  Value 0 indicate unlimited. </div> <div>  Helps the administrator to take control of the repository growth, and detect user's abuse. </div>
Advanced filters	<p>Enables search filtering on users and roles popup view.</p> <div>  Helps to find roles or user in a large list. </div>
Web skin	<p>Sets the default skin (css styles applied).</p> <div>  Info message </div>
Print preview	Enables or disables printing funtion.
Keywords enabled	Enables or disables keywords shortcuts.

Upload notify users	Shows or hides user notification on upload popup.
Notify external users	<p>Enables or disables send notifications to external users.</p> <div>  <i>By default the application is restricted to mail user's addresses.</i> </div> <div>  <i>For example, sending documents to an external mail address.</i> </div>
Browser plugin preview	When enabled, preview is based on browser pdf render capability.
Increase version	When enabled, users can choose between document upload as minor version or major version.
Extensions	Sets the enabled extensions for the profile.
Reports	Sets the enabled reports for the profile.
Workflows	Sets the available workflows for the profile.

Chat section

Field	Description
Enabled	Enables or disables chat.
Auto login	When enabled the user is automatically logged in chat as part of login process.




Wizard section

The Wizard only takes effect when uploading a new document, not when updating and existing.





In case an automation definition is present, these parameters has no effect.


The automation feature covers all these cases and more.

Field	Description
Metadata	Sets a metadata that will be required to be filled as part of uploading a new document.
Workflow	<p>Sets a workflows that will be executed as part of uploading a new document.</p> <div>  Some workflows require initial fields filled at the beginning, but not all. </div>
Keywords	<p>When enabled the application will ask user to add keywords as part of uploading a new document.</p> <div>  User can skip this step without adding any keyword. </div>
Categories	<p>When enabled the application will ask user to add categories as part of uploading a new document.</p> <div>  User can skip this step without adding any category. </div>

Pagination section

Pagination control

Field	Description
Enabled	Enables or disable pagination on file explorer view.
Page list	<p>Sets the distinct values in what will be shown in number of elements per page.</p> <div>  For example the value "10;25;50;100" set that maximum number of element per page can be 10, 25, 50 or 100. On the user view, the user can choose one of these options. </div> <div>  When pagination is enabled at least one value must be set To separate the distinct values must be used the character ";". </div>
Type filter	When enabled a filter by type is shown, this helps the user to show only documents, folders, records or mails at the same time. *

Misc filter	<p>Advanced filter helps on filtering by some fields:</p> <ul style="list-style-type: none"> • name • title • range of sizes • range of dates • author • version <div>  When extra metadata columns are show in file explorer, they are also available here. </div>
Show folders by default	By default shows or hides folders. *
Show documents by default	By default shows or hides document. *
Show mails by default	By default shows or hides mails. *
Show records by default	By default shows or hides records. *
<p>* When type filter is enabled the options show folders by default etc... they must be active, otherwise they are not taken in consideration and all types are shown.</p>	

Profile Components tab

Stacks section




<i>Field</i>	<i>Description</i>
<i>Taxonomy</i>	<i>Shows or hides taxonomy in the stack panel.</i>
<i>Categories</i>	<i>Shows or hides categories in the stack panel.</i>
<i>Thesaurus</i>	<i>Shows or hides the thesaurus in the stack panel.</i>
<i>Templates</i>	<i>Shows or hides the templates in the stack panel.</i>
<i>Personal</i>	<i>Shows or hides personal in the stack panel.</i>
<i>Mail</i>	<i>Shows or hides mail in the stack panel.</i>
<i>Metadata</i>	<i>Shows or hides metadata in the stack panel.</i>
<i>Trash</i>	<i>Shows or hides trash bin in the stack panel.</i>

Dashboard section

<i>Field</i>	<i>Description</i>
<i>User</i>	<i>Shows or hides the user panel on dashboard view.</i>
<i>Mail</i>	<i>Shows or hides the new mail panel in the dashboard view.</i>
<i>News</i>	<i>Shows or hides the news panel in the dashboard view.</i>
<i>General</i>	<i>Shows or hides the general panel in the dashboard view.</i>

Workflow	Shows or hides the workflow panel in the dashboard view.
Keywords	Shows or hides the tag cloud panel in the dashboard view.

Tab section

Field	Description
Default tab	<p>Sets the default tab visible when user login the application.</p> <div>  Available options: <ul style="list-style-type: none"> • Desktop • Search • Dashboard • Administration </div> <div>  Must set an enabled view. </div>
Desktop	Shows or hides the desktop view.
Search	Shows or hides the search view.
Dashboard	Shows or hides the dashboard view.
Administration	<p>Shows or hides the administration view.</p> <div>  Administration view is only visible to users with "ROLE_ADMIN". </div>

Tab document section

The tab document is visible when a document is selected in the desktop view.

Field	Description
Properties	Shows or hides the document properties tab.

Security	<i>Shows or hides the security tab.</i>
Notes	<i>Shows or hides the notes tab.</i>
Versions	<i>Shows or hides the versions tab.</i>
Version download	<i>Enables or disables the download document version.</i>
Preview	<i>Shows or hides the preview tab.</i>
Metadata	<i>Shows or hides the metadata tabs.</i>

Tab folder section

The tab folder is visible when a folder is selected in desktop view.

Field	Description
Properties	<i>Shows or hides the folder properties tab.</i>
Security	<i>Shows or hides the folder tab.</i>
Notes	<i>Shows or hides the notes tab.</i>

Tab mail section

Tab mail is visible when a mail is selected in desktop view.

Field	Description
Properties	<i>Shows or hides the mail properties tab.</i>
Security	<i>Shows or hides the mail tab.</i>
Notes	<i>Shows or hides the notes tab.</i>

<i>Preview</i>	<i>Shows or hides the preview tab.</i>
-----------------------	--

Profile Menu tab

These menu options have preference among toolbar options or contextual menu options.

Hierarchy:

- Menu
- Toolbar



Preference hierarchy. Upper hierarchy takes control over the lower. If the upper option is disabled then it will be also disabled at the lower one.

For example, when download menu option is disabled then it will not be shown at the toolbar although there will be enabled.


Menu section

These are the main menu options.




Field	Description
File	Shows or hides the file menu.
Edit	Shows or hides the edit menu.
Tools	Shows or hides the tools menu.
Bookmarks	Shows or hides the bookmark menu.
Templates	Shows or hides the templates menu.
Help	Shows or hides the help menu.

Menu bookmark section

Field	Description
Manage bookmarks	Shows or hides manage the bookmarks menu option.

Add bookmark	Show or hide add the bookmark menu option.
Set home	Shows or hides the set home menu option.
Go home	<p>Shows or hides the go to home menu option.</p> <div>  On loading, the application always try to open the default home. The default home is /okm:root that can be changed by end user. </div>


Menu tool section

Field	Description
Languages	Shows or hides the change language menu option.
Skin	Shows or hides the change skin menu option.
Debug	<p>Shows or hides the debug menu option.</p> <div>  When enabled is shown, it helps the administrator to find out possible UI bugs. </div>
Administration	<p>Shows or hides the administration menu option.</p> <div>  Open administration is on a new window. </div>
Preferences	Shows or hides the user preferences menu option.
Convert	<p>Shows or hides the convert menu option.</p> <div>  This option is used to convert documents to different formats. For example .doc to PDF. </div>

Menu help section

<i>Field</i>	<i>Description</i>
Documentation	Shows or hides the go to documentation website menu option.
Bug tracking	Shows or hides the go to bug tracking website menu option.
Support	Shows or hides the support menu option.
Forum	Shows or hides the go to public forum website menu option.
Changelog	Shows or hides the go to changelog menu option.
Web site	Shows or hides the go to openkm website menu option.
About	Shows or hides the application information menu option.

Menu file section

<i>Field</i>	<i>Description</i>
Create folder	Shows or hides the create folder menu option.
Find folders	Shows or hides the find folders menu option.
Find documents	Shows or hides the find documents menu option.
Similar documents	Shows or hides the find similar documents menu option.
Go folder	Shows or hides the go to folder menu option. <div>  This option for example is shown in the contextual menu at the search results. </div>
Download	Shows or hides the download menu option.

Download PDF	<i>Shows or hides the download as pdf menu option.</i>
Add document	<i>Shows or hides the add document menu option.</i>
Purge	<i>Shows or hides the purge menu option.</i>
Purge trash	<i>Shows or hides the purge all user trash menu option.</i>
Restore	<i>Shows or hides the restore from trash menu option.</i>
Start workflow	<i>Shows or hides the start workflow menu option.</i>
Refresh	<i>Shows or hides the refresh user interface menu option.</i>
Scanner	<i>Shows or hides the scanner tool menu option.</i>
Uploader	<i>Shows or hides the bulk upload menu option.</i>
Export	<i>Shows or hides the export to ZIP menu option.</i>
Create from template	<i>Shows or hides the create from template menu option.</i>
Send document link	<i>Shows or hides the send document link menu option.</i>
Send document attachment	<i>Shows or hides the send document attachment menu option.</i>
Forward e-mail	<i>Shows or hides the forward email menu option.</i>

Menu edit section

Field	Description

Lock	<i>Shows or hides the lock menu option.</i>
Unlock	<i>Shows or hides the unlock menu option.</i>
Update	<i>Shows or hides the update a document menu option.</i>
Edit	<i>Shows or hides the edit a document menu option.</i>
Cancel Edit	<i>Shows or hides the cancel edit a document menu option.</i>
Delete	<i>Shows or hides the delete menu option.</i>
Rename	<i>Shows or hides the rename menu option.</i>
Copy	<i>Shows or hides the copy menu option.</i>
Move	<i>Shows or hides the move menu option.</i>
Add subscription	<i>Shows or hides the add subscription menu option.</i>
Remove subscription	<i>Shows or hides the remove subscription menu option.</i>
Add metadata group	<i>Shows or hides the add metadata group menu option.</i>
Update metadata group	<i>Shows or hides the update metadata group menu option.</i>
Remove metadata group	<i>Shows or hides the remove metadata group menu option.</i>
Add note	<i>Shows or hides the add note menu option.</i>
Remove note	<i>Shows or hides the remove note menu option.</i>
Add category	<i>Shows or hides the add category menu option.</i>

<i>Remove category</i>	<i>Shows or hides the remove category menu option.</i>
<i>Add keyword</i>	<i>Shows or hides the add keyword menu option.</i>
<i>Remove keyword</i>	<i>Shows or hides the remove keyword menu option.</i>
<i>Merge PDF</i>	<i>Shows or hides the merge pdf documents menu option.</i>

Profile Other tab

Toolbar section

Set the toolbar buttons that are shown.



To enable the toolbar option it must also be enabled in the menu tab, otherwise it will not be visible in the toolbar.

Field	Description
Create folder	Shows or hides the create folder menu option.
Find folders	Shows or hides the find folders menu option.
Find documents	Shows or hides the find documents menu option.
Similar documents	Shows or hides the find similar documents menu option.
Download	Shows or hides the download document menu option.
Download PDF	Shows or hides the download document as pdf menu option.
Print	Shows or hides the print document menu option.
Lock	Shows or hides the lock menu option.
Unlock	Shows or hides the unlock menu option.
Add document	Shows or hides the add document menu option.
Edit	Shows or hides the edit a document menu option.
Update	Shows or hides the update a document menu option.

Cancel Edit	<i>Shows or hides the cancel edit a document menu option.</i>
Delete	<i>Shows or hides the delete menu option.</i>
Add metadata group	<i>Shows or hides the add metadata group menu option.</i>
Remove metadata group	<i>Shows or hides the remove metadata group menu option.</i>
Start workflow	<i>Shows or hides the start workflow menu option.</i>
Add subscription	<i>Shows or hides the add subscription menu option.</i>
Remove subscription	<i>Shows or hides the remove subscription menu option.</i>
Refresh	<i>Shows or hides the refresh user interface menu option.</i>
Home	<i>Shows or hides the go to home menu option.</i>
Scanner	<i>Shows or hides the scanner tool menu option.</i>
Uploader	<i>Shows or hides the bulk upload menu option.</i>
Splitter resize	<i>Shows or hides the splitter resize menu option.</i>

File browser section

Sets the file browser columns that are shown.


Field	Description
Status	<i>Shows or hides the status column.</i>
Massive	<i>Shows or hides the checkbox column.</i>

Icon	<i>Shows or hides the icon column.</i>
Name	<i>Shows or hides the name column.</i>
Size	<i>Shows or hides the size column.</i>
Last modified	<i>Shows or hides the last mofied date column.</i>
Author	<i>Shows or hides the author column.</i>
Version	<i>Shows or hides the version column.</i>
Column 1 to Column 10	<i>Shows or hides the column based on metadata value.</i>



Database query

This feature helps administrator accessing database data, easily and quickly without thirdparty database software client.

Allowed types:

Type	Description	List	Export
JDBC	SQL queries supported by the configured database.	When choosen is shown a list of available tables.	Yes.
Hibernate	Neutral database queries based on hibernate queries format.		No.
Metadata	Application specific format only for doing queries over OKM_DB_METADATA_VALUE. <div>  For more information see Creating your own database tables. </div>	When choosen is shown a list of available metadata tables.	Yes.
<ul style="list-style-type: none"> • Is allowed writing more than one query. • Each query must be written on a single line, break lines are not allowed. • Execute will process all written queries in same order has been written, top to bottom. Queries are executed as a FIFO (first in first out). • In case some query fails, that will not stop the execution of the others. 			

Execute JDBC query

- On bottom right **choose type JDBC**.
- Write some valid SQL.
- Click on **Execute button**.
- Optionaly the results can be exported to SQL or CSV
 - Click on  Export to sql icon.
 - Click on  Export to csv icon.

Example:

```
// nodeBase = documents + folders + mails + records
SELECT count(*) from OKM_NODE_BASE;
SELECT count(*) from OKM_NODE_DOCUMENT;
SELECT count(*) from OKM_NODE_FOLDER;
SELECT count(*) from OKM_NODE_MAIL;

// select first 10 records
SELECT * from OKM_NODE_BASE LIMIT 10;
SELECT * from OKM_NODE_DOCUMENT LIMIT 10;
SELECT * from OKM_NODE_DOCUMENT_VERSION LIMIT 10;
SELECT * from OKM_NODE_FOLDER LIMIT 10;
SELECT * from OKM_NODE_MAIL LIMIT 10;
```

Execute Hibernate query

- On bottom right **choose type Hibernate**.
- Write some valid HQL.
- Optionally you can check the "Show SQL" option.
- Click on **Execute button**.



When "Show SQL" option is checked is shown in results the real sql executed.

Example:

```
select count(*) from NodeBase;
select count(*) from NodeFolder;
```

Execute Metadata query

- On bottom right **choose type Hibernate**.
- Write some valid metadata query.
- Click on **Execute button**.
- Optionally the results can be exported to SQL or CSV
 - Click on Export to sql icon.
 - Click on Export to csv icon.

Example:

```
SELECT|country
```



Metadata country sample available at [Creating your own database tables](#).

Allowed sentences

Type	Format	Example
SELECT	SELECT TABLE SELECT TABLE FILTER	<i>SELECT country</i> <i>SELECT country \$country_name='Canada'</i> <i>SELECT country col01='Canada'</i>
UPDATE	UPDATE TABLE COLUMN UPDATE TABLE COLUMN FILTER	<i>UPDATE country \$country_name='France'</i> <i>UPDATE country \$country_name='France' \$country_id='001'</i>
DELETE	DELETE TABLE DELETE TABLE FILTER	<i>DELETE country</i> <i>DELETE country \$country_id='001'</i>
SENTENCE	SENTENCE TABLE1,...,TABLEn QUERY	<i>SENTENCE country select \$country_id, \$country_name from DatabaseMetadataValue dmv where dmv.table='country' order by \$country_name</i>
<ul style="list-style-type: none"> • Character \$ is used to indicate that the word is a column name. 		

Execute queries from sql file

- On bottom right **choose a valid sql file**.
- Click on **Import SQL script button**.

Reports

The application comes with Jasper reports engine. That helps administrator to create own reports.



The [JasperReports Library](#) is one of the world's most popular reporting engine. It is able to use data coming from any type of data source and produce pixel-perfect documents that can be viewed, printed or exported in a variety of document formats including HTML, PDF, Excel, OpenOffice and Word.



More information about how to build your own reports at [Creating your own reports](#).

More information about jasper report engine version that comes into application at [Application version compatibility table](#).



Keep on mind that any font used in the reports should be installed in the OpenKM server. For example, if you are using Arial (sample provided reports use this font) you should ensure Arial font is installed. For example, if you are using Ubuntu the package to install is **ttf-mscorefonts-installer**.


```
$ apt-get install ttf-mscorefonts-installer
```

In RedHat or CentOS the package is **msttcore-fonts**. You need to restart the server after this package is installed.


Reports are shown in a table structure.

Name	File name	Active
------	-----------	--------


Create a new report

- On the top right click on  **New report icon**.
- **Fill in the fields**.
- **Choose a report file** (*.rep or *.jrxml)
- Click on **Create button**.

Edit a report

- Click on  **Edit a report icon**.
- **Modify the fields**.
- **Optionally choose a report file** (*.rep or *.jrxml)
- Click on **Edit button**.

Delete a report

- Click on  **Delete a report icon.**
- Click on **Delete button.**

Show report report parameters







Optionally, reports can require parameters for execution.

For example a range of dates. These parameters will be shown as a form in the user interface when report will be executed.

- Click on  **Report params icon.**

Execute a report

- Click on  **Execute report icon.**
- **Fill in the fields** (when report comes with parameters).
- Choose report output format
 -  **PDF format.**
 -  **RTF format.**
 -  **CSV format.**

Reports pack



Most of these reports are based on Activity log events audit trail.

For more information [Activity log](#).

Template reports

Name	Use Activity log	Require parameters	Description
JavaReportPattern.jrxml	No.	No.	Template pattern that use Java to retrieve data.
SQLReportPattern.jrxml	No.	No.	Template pattern that use SQL to retrieve data.




Authentication

Name	Use Activity log	Require parameters	Description
-	Yes.	Yes.	Users login, filtered by date range.
UsersLogout.rep	Yes.	Yes.	Users logout, filtered by date range.
UsersSessionExpiration.rep	Yes.	Yes.	Users that session expired, filtered by date range.

Documents

Name	Use Activity log	Require parameters	Description
DocumentCanceledCheckout.rep	Yes.	Yes.	Lists documents that cancelled the edition, filtered by date range.
DocumentCheckin.rep	Yes.	Yes.	List documents updated, filtered by date range.
DocumentCheckout.rep	Yes.	Yes.	List documents on edition, filtered by date


			range.
<u>DocumentCreate.rep</u>	Yes.	Yes.	List documents created, filtered by date range.
<u>DocumentDelete.rep</u>	Yes.	Yes.	List documents deleted, filtered by date range.
<u>DocumentGetChildren.rep</u>	Yes.	Yes.	<p>Show when the users listed the documents into some folder or record. Results are filtered by date range.</p> <div>  To list documents into a folder is called internally method named "getChildren". </div>
<u>DocumentContent.rep</u>	Yes.	Yes.	<p>Show when the users retrieved the content of a document. Results are filtered by date range.</p> <div>  For example, when a user download or preview a document (among other actions) is retrieving the content of the document. </div>
<u>DocumentContentByVersion.rep</u>	Yes.	Yes.	Show when the users retrieved a specific version of the content of a document. Results are filtered by date range.
<u>DocumentGetProperties.rep</u>	Yes.	Yes.	<p>Show when user retrieved the properties of a document. Results are filtered by date range.</p> <div>  For example, when document is selected, is called internally a method named "getProperties" to fill the contents of properties tab in user interface. </div>
<u>DocumentGetVersionHistory.rep</u>	Yes.	Yes.	Shown when the users retrieved the list of all

			versions of a document. Results are filtered by date range.
DocumentGetPropertyGroupProperties.rep	Yes.	Yes.	<p>List when has been retrieved the fields of a metadata group.</p> <div>  <p>For example, when a metadata group is shown in users interface, is called internally a method named "getPropertyGroupProperties" to fill the metadata group values.</p> </div>
DocumentLock.rep	Yes.	Yes.	List documents updated, filtered by date range.
DocumentMove.rep	Yes.	Yes.	List documents moved, filtered by date range.
DocumentPurge.rep	Yes.	Yes.	<p>List documents purged, filtered by date range.</p> <div>  <p>When a document is purged it will definitively disappear from application repository.</p> </div>
DocumentRenamed.rep	Yes.	Yes.	List documents renamed, filtered by date range.
DocumentUnlock.rep	Yes.	Yes.	List documents unlocked, filtered by date range.
DocumentsAccessibleByUserOrdered.jrxml	No.	No.	<p>Retrieve a list of available documents - security filtered - ordered by folder name.</p> <div>  <p>Optimized for csv format. Not recommended for large repositories, can take a lot of time.</p> </div>
DuplicatedFiles.jrxml	No.	No.	List of duplicated files into application



			repository, based on document checksum.
ListDocumentsWithChecksum.rep	No.	Si.	List of all documents into some folders and subfolders with checksum value.


Folders

Name	Use Activity log	Require parameters	Description
FolderCopy.rep	Yes.	Yes.	List folders copied, filtered by date range.
FolderCreate.rep	Yes.	Yes.	List folders created, filtered by date range.
FolderDelete.rep	Yes.	Yes.	List folders deleted, filtered by date range.
FolderGetChilden.rep	Yes.	Yes.	<p>Show when the users listed the folders into some folder or record. Results are filtered by date range.</p> <div>  To list folders into a folder is called internally method named "getChildren". </div>
FolderGetContentInfo.rep	Yes.	Yes.	<p>Shown when the internal method named "getContentInfo" has been executed.</p> <div>  To do some operations is needed to calculate folder deep, for example before starting folder ZIP download. There's an internal method named "getCotentInfo" that does it. </div>
FolderGetProperties.rep	Yes.	Yes.	<p>Show when user retrieved the properties of a folder. Results are filtered by date range.</p> <div>  For example, when folder is selected, is called internally a method named "getProperties" to fill the contents of properties tab in user interface. </div>

FolderMove.rep	Yes.	Yes.	List folders moved, filtered by date range.
FolderPurge.rep	Yes.	Yes.	List folders purged, filtered by date range. <div>  When a folder is purged it will definitively disappear from application repository. </div>
FolderRename.rep	Yes.	Yes.	List folders renamed, filtered by date range.

Mails

Name	Use Activity log	Require parameters	Description
MailCreate.rep	Yes.	Yes.	List mails created, filtered by date range.
MailGetProperties.rep	Yes.	Yes.	Shown when a user retrieved the properties of a mail. Results are filtered by date range. <div>  For example, when mail is selected, is called internally a method named "getProperties" to fill the contents of properties tab in user interface. </div>
MailPurge.rep	Yes.	Yes.	List mails purged, filtered by date range. <div>  When a mail is purged it will definitively disappear from application repository. </div>
MailRename.rep	Yes.	Yes.	List mails renamed, filtered by date range.
MailMove.rep	Yes.	Yes.	List mails moved, filtered by date range.
MailCopy.rep	Yes.	Yes.	List mails copied, filtered by date range.

MailGetChilden.rep	Yes.	Yes.	<p>Show when the users listed the mails into some folder or record.</p> <p>Results are filtered by date range.</p> <div>  <p>To list mails into a folder is called internally method named "getChildren".</p> </div>
------------------------------------	------	------	---

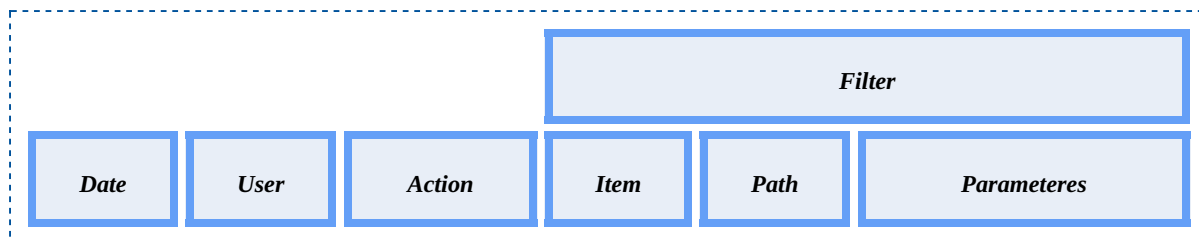
Others

Name	Use Activity log	Require parameters	Description
WorkflowWorkload.jrxml	No.	No.	Each workflow task with the time spent on it.
RegisteredUsers.jrxml	No.	No.	Registered users.

Activity log

This application performs a complete audit trail. Audit trail is mainly a datastore service. It defines a data record structure that will be used for storing audit information.

The activity log is shown in a table structure.



Filter options:



Field	Description	Required
From date	Start of date range.	true.
To date	End of date range.	true.
User	Specific user.	false.
Action	Any action.	false.



Auth actions

Field / Property	Description
LOGIN	User log in.
LOGOUT	User log out.
SESSION_EXPIRATION	User session expiration date
GRANT_USER	Adds user grant on a node.
REVOKE_USER	Removes user grant on a node.

GRANT_ROLE	<i>Add role grant on a node.</i>
REVOKE_ROLE	<i>Removes user grant on a node.</i>



Document actions

<i>Field / Property</i>	<i>Description</i>
CANCEL_DOCUMENT_CHECKOUT	<i>Cancels document edition.</i>
CHECKIN_DOCUMENT	<i>Updates document.</i>
CHECKOUT_DOCUMENT	<i>Edits document.</i>
CREATE_DOCUMENT	<i>Creates a new document.</i>
DELETE_DOCUMENT	<i>Deletes a document.</i>
GET_CHILDREN_DOCUMENTS	<i>Retrieves a document list in a node.</i> <div>  <i>List of documents in a folder or record.</i> </div>
GET_DOCUMENT_CONTENT	<i>Retrieves binary data.</i>
GET_DOCUMENT_CONTENT_BY_VERSION	<i>Retrieves binary data by document version.</i>
GET_DOCUMENT_PROPERTIES	<i>Retrieves all document properties.</i> <div>  <i>It does not include metadata. It has a direct relationship with the panel properties when a document is selected.</i> </div>
GET_DOCUMENT_VERSION_HISTORY	<i>Retrieves a list of all versions of a document.</i>



GET_PROPERTY_GROUP_PROPERTIES	<i>Retrieves all metadata properties of a document.</i>
LOCK_DOCUMENT	<i>Locks a document.</i>
MOVE_DOCUMENT	<i>Moves a document.</i>
PURGE_DOCUMENT	<i>Purges a document.</i> <div>  <i>A purged document is definitively removed from the application.</i> </div>
RENAME_DOCUMENT	<i>Renames a document.</i>
SET_DOCUMENT_PROPERTIES	<i>Sets document properties.</i> <div>  <i>For example when a user is subscribed to a document.</i> </div>
UNLOCK_DOCUMENT	<i>Unlocks a document</i>


Folder actions

Field / Property	Description
COPY_FOLDER	<i>Copies a folder.</i>
CREATE_FOLDER	<i>Creates a folder.</i>
DELETE_FOLDER	<i>Deletes a folder.</i>
GET_CHILDREN_FOLDERS	<i>Retrieves a folder list in a node.</i> <div>  <i>List of folders in a folder or record.</i> </div>

GET_FOLDER_CONTENT_INFO	Counts nodes into a folder (documents, folders, records and mails that are into).
GET_FOLDER_PROPERTIES	<p>Retrieves all folder properties.</p> <div>  Does not include metadata. it has a direct relationship with panel properties when a folder is selected. </div>
MOVE_FOLDER	Moves a folder.
PURGE_FOLDER	<p>Purges a folder.</p> <div>  A purged folder is definitively removed from application. </div>
RENAME_FOLDER	Renames a folder.

Mail actions

Field / Property	Description
CREATE_MAIL	Creates a mail.
GET_MAIL_PROPERTIES	<p>Retrieves all mail properties.</p> <div>  Does not include metadata. It has a direct relationship with panel properties when a mail is selected. </div>
DELETE_MAIL	Deletes a mail.
PURGE_MAIL	<p>Purges a mail.</p> <div>  A purged mail is definitively removed from application. </div>
RENAME_MAIL	Renames a mail.

MOVE_MAIL	Moves a mail.
COPY_MAIL	Copies a mail.
GET_CHILDREN_MAILS	Retrieves mail list in some node. <div> List of mail in a folder or record.</div>
DOWNLOAD_MAIL	Downloads a mail


Repository actions

Field / Property	Description
PURGE_TRASH	Trash is purged. <div>All related nodes are definitively removed from application.</div>

Admin actions

 These actions are only done by users with Administration grants.
--

Field / Property	Description
ADMIN_ACTIVITY_LOG	The administrator uses "activity log".
ADMIN_ACTIVE_SESSIONS	The administrator uses Logged users .
ADMIN_USER_CREATE	Creates a user.
ADMIN_USER_EDIT	Edits a user.
ADMIN_USER_DELETE	Deletes a user.

ADMIN_USER_ACTIVE	<i>Enables or disables a user.</i>
ADMIN_ROLE_CREATE	<i>Creates a role.</i>
ADMIN_ROLE_EDIT	<i>Edits a role.</i>
ADMIN_ROLE_DELETE	<i>Deletes a role.</i>
ADMIN_ROLE_ACTIVE	<i>Enables or disables a role.</i>
ADMIN_CHECK_EMAIL	<p><i>Checks email poll.</i></p> <div>  <p><i>The application reads email accounts to import emails.</i></p> </div>
ADMIN_CONFIG_CREATE	<i>Creates a configuration parameters.</i>
ADMIN_CONFIG_EDIT	<i>Edits a configuration parameter.</i>
ADMIN_CONFIG_DELETE	<i>Deletes a configuration parameter.</i>
ADMIN_CRONTAB_CREATE	<i>Creates a crontab task.</i>
ADMIN_CRONTAB_EDIT	<i>Edits a contab task.</i>
ADMIN_CRONTAB_DELETE	<i>Deletes a crontab task.</i>
ADMIN_CRONTAB_EXECUTE	<i>Executes a crontab task.</i>
ADMIN_DATABASE_QUERY	<i>Executes a database query.</i>
ADMIN_DATABASE_UPDATE	<i>Executes an update database query.</i>

ADMIN_LANGUAGE_CREATE	<i>Creates a new language.</i>
ADMIN_LANGUAGE_EDIT	<i>Edits a language.</i>
ADMIN_LANGUAGE_DELETE	<i>Deletes a language.</i>
ADMIN_LANGUAGE_IMPORT	<i>Imports a language.</i>
ADMIN_LOGCAT_LIST	<i>The administrator user logcat tool.</i>
ADMIN_LOGCAT_VIEW	<i>The administrator user logcat tool.</i>
ADMIN_LOGGED_USERS	<i>The administrator uses Logged users.</i>
ADMIN_MAIL_ACCOUNT_CREATE	<i>Creates a mail account.</i>
ADMIN_MAIL_ACCOUNT_EDIT	<i>Edits a mail account.</i>
ADMIN_MAIL_ACCOUNT_DELETE	<i>Deletes a mail account.</i>
ADMIN_MAIL_ACCOUNT_CHECK	<i>Checks a mail account.</i>
ADMIN_MAIL_FILTER_CREATE	<i>Creates a mail account filter.</i>
ADMIN_MAIL_FILTER_EDIT	<i>Edits a mail account filter.</i>
ADMIN_MAIL_FILTER_DELETE	<i>Deletes a mail account filter.</i>
ADMIN_MAIL_FILTER_RULE_CREATE	<i>Creates a mail account rule.</i>
ADMIN_MAIL_FILTER_RULE_EDIT	<i>Edits a mail account rule.</i>
ADMIN_MAIL_FILTER_RULE_DELETE	<i>Delete a mail account rule.</i>


ADMIN_MIME_TYPE_CREATE	<i>Creates a new MIME type</i>
ADMIN_MIME_TYPE_EDIT	<i>Edits MIME type.</i>
ADMIN_MIME_TYPE_DELETE	<i>Deletes MIME type.</i>
ADMIN_USER_PROFILE_CREATE	<i>Creates a profile.</i>
ADMIN_USER_PROFILE_EDIT	<i>Edits a profile.</i>
ADMIN_USER_PROFILE_DELETE	<i>Deletes a profile.</i>
ADMIN_PROPERTY_GROUP_REGISTER	<i>Registers a metadata group.</i>
ADMIN_PROPERTY_GROUP_LIST	<i>Lists metadata groups.</i>
ADMIN_REPORT_CREATE	<i>Creates a report.</i>
ADMIN_REPORT_EDIT	<i>Edits a report.</i>
ADMIN_REPORT_DELETE	<i>Deletes a report.</i>
ADMIN_REPORT_EXECUTE	<i>Executes a report.</i>
ADMIN_REPOSITORY_SEARCH	<i>Performes a Lucene search.</i>
ADMIN_REPOSITORY_REINDEX	<i>Reindexes Lucene repository.</i>
ADMIN_REPOSITORY_UNLOCK	<i>Unlocks a node from a repository view.</i>
ADMIN_REPOSITORY_CHECKIN	<i>Updates a node from a repository view.</i>

ADMIN_REPOSITORY_EDIT	<i>Edits a node from a repository view.</i>
ADMIN_REPOSITORY_SAVE	<i>Modify a node from a repository view.</i>
ADMIN_REPOSITORY_LIST	<i>Refreshes folder contents from repository view.</i>
ADMIN_REPOSITORY_REMOVE_CONTENT	<i>Purges folder lo dentro</i>
ADMIN_REPOSITORY_REMOVE_CURRENT	<i>Deletes all folder todo</i>
ADMIN_WORKFLOW_REGISTER	<i>Registers a workflow.</i>
ADMIN_PROCESS_DEFINITION_DELETE	<i>Deletes a workflow process definition.</i>
ADMIN_PROCESS_INSTANCE_DELETE	<i>Deletes a workflow process instance.</i>
ADMIN_PROCESS_INSTANCE_END	<i>Workflow process instance finished.</i>
ADMIN_PROCESS_INSTANCE_RESUME	<i>Workflow process instance resumed.</i>
ADMIN_PROCESS_INSTANCE_SUSPEND	<i>Workflow process instance suspended.</i>
ADMIN_PROCESS_INSTANCE_ADD_COMMENT	<i>Adds comment on a workflow process instance.</i>
ADMIN_PROCESS_INSTANCE_VARIABLE_DELETE	<i>Deletes variable on a workflow process instance.</i>
ADMIN_PROCESS_INSTANCE_VARIABLE_ADD	<i>Adds variable on a workflow process instance.</i>
ADMIN_TASK_INSTANCE_SET_ACTOR	<i>Sets actor on a a workflow task instance.</i>
ADMIN_TASK_INSTANCE_START	<i>Workflow task instance started.</i>

ADMIN_TASK_INSTANCE_END	<i>Workflow task instance ended.</i>
ADMIN_TASK_INSTANCE_SUSPEND	<i>Workflow task instance suspended.</i>
ADMIN_TASK_INSTANCE_ADD_COMMENT	<i>Adds comment on a workflow task instance.</i>
ADMIN_TASK_INSTANCE_VARIABLE_DELETE	<i>Deletes variable on a workflow task instance.</i>
ADMIN_TASK_INSTANCE_VARIABLE_ADD	<i>Adds variable on a workflow task instance.</i>
ADMIN_TASK_INSTANCE_RESUME	<i>Resumes a workflow task instance.</i>
ADMIN_TOKEN_SIGNAL	<i>Workflow task signal sent from administration view.</i>
ADMIN_TWITTER_ACCOUNT_CREATE	<i>Creates a twitter account.</i>
ADMIN_TWITTER_ACCOUNT_EDIT	<i>Edits a twitter account.</i>
ADMIN_TWITTER_ACCOUNT_DELETE	<i>Deletes a twitter account.</i>
ADMIN_USER_CONFIG_EDIT	<i>Changes a user profile.</i>
ADMIN_SCRIPTING	<i>Executes scripting.</i>

Misc actions

<i>Field / Property</i>	<i>Description</i>
MISC_OPENKM_START	<i>Starts application.</i>
MISC_OPENKM_STOP	<i>Stops application.</i>
	<i>Application check.</i>

<i>MISC_STATUS</i>	<div> <i>Several internal checks to ensure application is alive:</i><ul style="list-style-type: none">• <i>Database connection.</i>• <i>Repository connection.</i></div>
<i>MISC_TEXT_EXTRACTION_FAILURE</i>	<i>Text extraction failure.</i>





Workflow

In Workflow view are shown the registered workflows. In this view the user can register new workflows and view or modify existing workflow process instances.

Workflow are shown in a table structure.


Process ID	Process Name	Version	Action
------------	--------------	---------	--------

Description:

Property	Description
Process ID	Each workflow has an unique Process ID.
Process Name	<p>The workflow Process Name is set into the workflow process definition.</p> <div>  <p>The workflow process definition is an XML. In the first tag is set the Process Name.</p> <pre><?xml version="1.0" encoding="UTF-8"?> <process-definition xmlns="urn:jbpml.org:jpd1-3.2" name="purchase"> ... </process-definition></pre> </div>
Version	<p>Workflow version.</p> <div>  <p>When is a new workflow is registered, if exists a previous workflow with the same Process Name, then the version will be increased in one unit.</p> </div>
Actions	<p>Available actions:</p> <ul style="list-style-type: none">  View (manage process instances, form definition, process image).  Delete

Register workflow

- On the right bottom, select the **.par file** (workflow file type).

- Click  **Register process definition button.**

Workflow forms definition

Workflow forms are based on formal XML definition. The DTD (Document Type Definition) defines the structure and the legal elements and attributes of an XML document.



You can see the latest DTD in our website <http://www.openkm.com/dtd/workflow-forms-{major}.{minor}.dtd> where {major} and {minor} values are two numbers (major release and minor release). For example the location for version 2.6 of the dtd is <http://www.openkm.com/dtd/workflow-forms-2.4.dtd>



Each application version supports only a couple of DTD definitions, read: [Application version compatibility table](#) to see which are supported by your application.

Basic XML skeleton

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
"http://www.openkm.com/dtd/workflow-forms-2.4.dtd">
<workflow-forms>
</workflow-forms>
```

- The **DOCTYPE** is a formal definition of some DTD. In the previous example the **workflow-forms-2.4.dtd**.
- The XML begins with **<workflow-forms>** tag and ends with **</workflow-forms>**. Into this hierarchy are defined the metadata groups.

Workflow task form definition

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
"http://www.openkm.com/dtd/workflow-forms-2.4.dtd">
<workflow-forms>
  <workflow-form task="evaluate price">
  </workflow-form>
</workflow-forms>
```

- Each **<workflow-form> ... </workflow-form>** block identify the **beginning and end** of new workflow form task definition.
- Each task has a **unique name**, "evaluate price".
- Must not use special characters for name.

Attribute	Description	Required
name	Unique field identifier.	true



Two metadata groups can't have the same name. **Name must be unique.**

The name must start with "okg:".

Must not use special character for task name. We suggest (0-9a-zA-Z_).

Basic example

Available field types:

- Checkbox
- Button
- Download
- Input
- Select
- Separator
- Suggestbox
- Text
- TextArea
- Upload

More information about field type elements at [Workflow fields](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
    "http://www.openkm.com/dtd/workflow-forms-2.4.dtd">
<workflow-forms>
  <workflow-form task="evaluate price">
    <input label="Purchase price" name="price" data="price" readonly="true" />
    <textarea label="Purchase description" name="description" data="description" read
    <button name="approve" label="Approve" transition="approve"/>
    <button name="deny" label="Deny" transition="deny"/>
  </workflow-form>
</workflow-forms>
```

Using file system DTD

If the server has no access to the Internet, then the DTD file must be accessible in some way, for example can store a copy in your application server.



The XML definition must be changed to refer to the new DTD location, otherwise you will get an application error while trying to process a DTD file.

It can be a file system (example below) or a valid intranet URL.

```
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.4//EN"
```

```
"file:///home/openkm/workflow-forms-2.4.dtd">
```

- In the example, **/home/openkm/workflow-forms-2.4.dtd** is the file system path to the file.

Workflow fields

These are the available field types:

Field type	Description
Checkbox	Workflow Checkbox field
Button	Workflow Button field
Download	Workflow Download field
Input	Workflow Input field
Print	Workflow Print field
Select	Workflow Select field
Separator	Workflow Separator field
Suggestbox	Workflow Suggestbox field
Text	Workflow Text field
TextArea	Workflow TextArea field
Upload	Workflow Upload field

Some field types can have children elements, validator or option elements. This hierarchy is permitted:

- workflow form
 - Checkbox
 - Validator
 - Button
 - Download

- *Node*
 - *Validator*
- *Input*
 - *Validator*
- *Print*
 - *Node*
 - *Validator*
- *Select*
 - *Option*
 - *Validator*
- *Separator*
- *Suggestbox*
 - *Validator*
- *Text*
- *TextArea*
 - *Validator*
- *Upload*
 - *Validator*

Workflow Button field

Buttons are typically used by users - in user interface - for decision making.



For example two buttons, accept and deny. One for accept something and the other one to deny.



Attribute	Description	Required
label	The text shown as label in user interface.	true
name	<p>A unique field identifier</p> <div> Two fields can't have the same name into the same task definion. The Name must be unique. </div>	true
transition	If present, a process instance token will follow the indicated transition.	false
confirmation	The button will show as a popup with an operation confirmation message. The message text used is confirmation value.	false
validate	Default value is "true".	false
style	<p>Icon style applied to the button. Possible values:</p> <ul style="list-style-type: none"> • yes • no • add • delete • download • downloadZip • home • view • change • compact • clean • search • save 	false

	<ul style="list-style-type: none">• <i>comment</i>	
width	<i>The width of the HTML element.</i>	<i>false</i>
height	<i>The height of the HTML element.</i>	<i>false</i>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
    "http://www.openkm.com/dtd/workflow-forms-2.4.dtd">
<workflow-forms>
  <workflow-form task="evaluate price">
    <button name="deny" label="Deny" transition="deny"/>
  </workflow-form>
</workflow-forms>
```

Workflow Download field

This workflow is used to download a document.

Attribute	Description	Required
label	The text shown as a label in user interface.	true
name	Unique name identifier  Two fields can't have the same name in the same task definion. Name must be unique.	true
width	The width of the HTML element.	false
height	The height of the HTML element.	false
data	When present, it is an identifier used to load data.  For example to dynamically load the uuid or path of the node to download.	false
Child elements: <ul style="list-style-type: none"> • Node. For more information see Workflow Node element. • Validator. For more information see Workflow Validator element. 		

Basic download example

- Download button with node element.
- **uuid** is a valid document uuid.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
    "http://www.openkm.com/dtd/workflow-forms-2.4.dtd" [
  <workflow-forms>
    <workflow-form task="download test">
      <download name="download" label="Download document">
        <node label="download" uuid="afb8ac33-912d-4515-e29b-1c01b693809a" />
      </download>
    </workflow-form>
  </workflow-forms>
```

Dynamic download example

- Download button with node element.
- **uuid** is a valid document uuid.
- **data** is a valid mapping variable. This **variable** is used to dynamically set **uuid attribute** into node element.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
    "http://www.openkm.com/dtd/workflow-forms-2.4.dtd">
<workflow-forms>
  <workflow-form task="download test">
    <download name="download" label="Download document" data="dynamic">
      <node label="download" />
    </download>
  </workflow-form>
</workflow-forms>
```

Workflow Validator element

The goal of the *Validation* element is to ensure that the user provides necessary and properly formatted information needed to successfully complete an operation.

Validator can only be used in combination with other fields:

- [Workflow Input field](#)
- [Workflow Suggestbox field](#)
- [Workflow Upload field](#)
- [Workflow Download field](#)
- [Workflow Print field](#)
- [Workflow Checkbox field](#)
- [Workflow Text field](#)
- [Workflow TextArea field](#)
- [Workflow Select field](#)

Available validators:

<i>name</i>	<i>Description</i>
req	Required value.
alpha	Alphanumeric value.
dec	Decimal value.
num	Numeric value.
email	E-mail value.
url	URL value.
maxlen	Sets the maximum length allowed.
minlen	Sets the minimum length required.

lt	Less than.
gt	Greater than.
min	Minimum value.
max	Maximum value.
regexp	Regular expression.
Parent: <ul style="list-style-type: none"> • Workflow Input field • Workflow Suggestbox field • Workflow Upload field • Workflow Download field • Workflow Print field • Workflow Checkbox field • Workflow Text field • Workflow TextArea field • Workflow Select field 	

Basic example

- Value required "**req**".
- Value must be decimal "**dec**".
- The maximum length of the decimal value is 6 digits "**maxlen**".

```
<input label="Decimal value" name="decimal">
  <validator type="dec"/>
  <validator type="req"/>
  <validator type="maxlen" parameter="6"/>
</input>
```

Regular expression

If you want a validation which is not included in these predefined, you can use a regular expression. For example, imagine you

want to restrict an input to lowercase ASCII values:

```
<input label="Lowercase value" name="regExp">  
  <validator type="regexp" parameter="[a-z]+" />  
</input>
```

Workflow Node element

This option element is only used in combination with:

- [Workflow Download field](#)
- [Workflow Print field](#)

The option value represents a list of distinct values in a Selected field.

Attribute	Description	Required
label	The text shown as a label in the user interface.	true
path	Node path. Usually a document path.	false
uuid	Node uuid. Usually a document valid uuid.	false
Parent : <ul style="list-style-type: none"> • Workflow Download field • Workflow Print field 		

Basic download example

- Download button with node element.
- **uuid** is a valid document uuid.

```
<workflow-form task="download test">
  <download name="download" label="Download document">
    <node label="download" uuid="afb8ac33-912d-4515-e29b-1c01b693809a" />
  </download>
</workflow-form>
```






Take it as an example. Normally it wants to assign dynamically a uuid, for it is used data attribute.

```
<download name="download" label="Download document" data="dynamic" />
```

Workflow Upload field

Upload is used to upload a document.

Attribute	Description	Required
label	The text shown as a label in user interface.	true
name	<p>Unique name identifier</p> <div>  <p>Two fields can't have the same name in the same task definion. Name must be unique.</p> </div>	true
width	The width of the HTML element.	false
height	The height of the HTML element.	false
data	<p>When present, is an identifier used to load data.</p> <div>  <p>For example dynamically set the folderPath or folderUuid.</p> </div>	false
type	<p>Values:</p> <ul style="list-style-type: none"> • create (the operation will be create new document) • update (the operation will be update existing document) <div>  <p>When it is not present, the default value is set to create.</p> </div>	false
folderPath	The folder path where the document will be stored.	false
folderUuid	The folder uuid where the document will be stored.	false
documentName	Forced name of the uploaded document.	false
documentUuid	The uuid of the document to be updated.	false



To upload a new document, you should set the `folderPath` or `folderUuid`, optionally `documentName` if we want to force the document name.

To update an existing document, you should set `documentUuid` and put `type` to `update`.

Child elements:

- **Validator.** For more information see [Workflow Validator element](#).

Basic upload example

- Upload a new document to a folder.
- **folderUuid** is a valid folder uuid. You can also use **folderPath** and set a path, which is more human readable.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
    "http://www.openkm.com/dtd/workflow-forms-2.4.dtd">
<workflow-forms>
  <workflow-form task="upload test">
    <upload name="upload" label="Upload document" folderUuid="fca2d85e-0e01-418c" />
  </workflow-form>
</workflow-forms>
```

Document update

- Create a new version of a given document.
- **documentUuid** is the UUID of the document to be replaced.
- Note the **type** parameter which is set to `update`.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
    "http://www.openkm.com/dtd/workflow-forms-2.4.dtd">
<workflow-forms>
  <workflow-form task="upload test">
    <upload name="upload" label="Upload document" type="update" documentUuid="e6a0630" />
  </workflow-form>
</workflow-forms>
```

Upload example with dynamic parameters

- Upload a document into a non-predefined folder.
- **data** is a valid mapping variable. This variable is used to dynamically set `folderPath` attribute.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
    "http://www.openkm.com/dtd/workflow-forms-2.4.dtd">
<workflow-forms>
  <workflow-form task="upload test">
```

```
<upload name="upload" label="Upload document" data="updData" />
</workflow-form>
</workflow-forms>
```

This is the Java code which set the data and should be executed before the task which displays the form element:

```
import org.jbpm.graph.def.ActionHandler;
import org.jbpm.graph.exe.ExecutionContext;

import com.openkm.bean.form.Upload;



public class UploadAction implements ActionHandler {
    private static final long serialVersionUID = 1L;

    @Override
    public void execute(ExecutionContext ctx) throws Exception {
        Upload upd = new Upload();
        upd.setFolderPath("/okm:root/dstFld");
        ctx.setVariable("updData", upd);

        // Go to next node
        ctx.getToken().signal();
    }
}
```

Workflow Print field

Used to print a document.

Attribute	Description	Required
label	The text shown as label in user interface.	true
name	Unique name identifier  Two fields can't have the same name in the same task definion. Name must be unique.	true
width	The width of the HTML element.	false
height	The height of the HTML element.	false
data	When present, is an identifier used to load data.  For example dynamically set the uuid or path of the document to download.	false
Child elements: <ul style="list-style-type: none"> • Node. For more information Workflow Node element. • Validator. For more information Workflow Validator element. 		

Basic print example

- Print a document
- **uuid** is a valid document uuid.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
    "http://www.openkm.com/dtd/workflow-forms-2.4.dtd" [
  <workflow-forms>
    <workflow-form task="print test">
      <print name="print" label="Print document">
        <node label="print" uuid="afb8ac33-912d-4515-e29b-1c01b693809a" />
      </print>
    </workflow-form>
  </workflow-forms>
```

Dynamic print example



- Print an existing document
- **data** is a mapping valid variable. This **variable** is **used to dynamically set uuid attribute** into the node.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
    "http://www.openkm.com/dtd/workflow-forms-2.4.dtd">
<workflow-forms>
  <workflow-form task="print test">
    <print name="print" label="Print document" data="dynamic">
      <node label="print" />
    </print>
  </workflow-form>
</workflow-forms>
```

Workflow Checkbox field

Checkbox




Best used to represent boolean data:

Attribute	Description	Required
label	The text shown as label in user interface.	true
name	Unique field identifier  Two fields can't have the same name in the same task definion. Name must be unique.	true
width	The width of the HTML element.	false
height	The height of the HTML element.	false
readonly	Set field, it will not be able to be modified in user interface. By default the value is "false".	false
data	When present, is an identifier used to load data.  For example to dynamically load the true or false value of the checkbox.	
Child elements: <ul style="list-style-type: none"> Validator. For more information see Workflow Validator element. 		

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
    "http://www.openkm.com/dtd/workflow-forms-2.4.dtd">
<workflow-forms>
  <workflow-form task="checkbox test">
    <checkbox label="Checkbox label" name="checkbox" />
  </workflow-form>
</workflow-forms>
```


Workflow Input field

In most cases is used to enter free form text, although can be used to save http links, date data or refer to application folder (as internal link to this folder).

Attribute	Description	Required
label	The text shown as a label in the user interface.	true
name	Unique field identifier. <div>  Two fields can't have the same name in the same task definition. Name must be unique. </div>	true
type	Type value. Allowed types: <ul style="list-style-type: none"> • text. • date. • link. • folder. 	true
value	Can specify a default value for this component. <div>  There's a special case for date types. Data is stored in format "yyyyMMddHHmmss", take care of it when you're using OpenKM API. </div>	false
width	The width of the HTML element.	false
height	The height of the HTML element.	false
data	When present, is an identifier used to load data. <div>  For example to dynamically load the Input value. </div>	
readonly	Set field will not be able to be modified in user interface. By default the value is "false".	false

Child elements:

- *Validator.* For more information see [Workflow Validator element](#).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
    "http://www.openkm.com/dtd/workflow-forms-2.4.dtd">
<workflow-forms>
  <workflow-form task="input test">
    <input label="Input label" name="input" />
  </workflow-form>
</workflow-forms>
```

Workflow Select field

Attribute	Description	Required
label	The text shown as label in user interface.	true
name	<p>Unique field identifier.</p> <div>  <p>Two fields can't have the same name in the same task definion. Name must be unique.</p> </div>	true
type	<p>Type value. Allowed types:</p> <ul style="list-style-type: none"> • simple (single choice allowed). • multiple (multiple choice allowed). 	true
table	<p>Metadata table name.</p> <div>  <p>Application can store values into metadata table, that can be used in this query.</p> </div>	*
width	The width of the HTML element.	false
height	The height of the HTML element.	false
optionsQuery	A metadata query or SQL query used to get the select options.	*
data	When present, is an identifier used to load data.	false
optionsData	When present, is an identifier used to load options.	false
readonly	Set field will not be able to be modified in user interface. By default the value is " false ".	false
Child elements:		

- Option. Couples of XML `<option label="label text" value="unique identifier value"/>` values or `<option label="label text" value="unique identifier value" parentValue="parent unique identifier value"/>` in case is set some hierarchical relationship.
- Validator. For more information see [Workflow Validator element](#).

* There're two options to get option values:

- Directly defined in XML.
- Metadata table (table + optionsQuery).



One of these options must be defined, otherwise you will get an error. A combination of several of these cannot be done, only one can be chosen.

Basic select example

- Type is equal to **simple**.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
    "http://www.openkm.com/dtd/workflow-forms-2.4.dtd">
<workflow-forms>
  <workflow-form task="select test">
    <select label="select label" name="select" type="simple">
      <option label="one" value="001" />
      <option label="two" value="002" />
      <option label="three" value="003" />
    </select>
  </workflow-form>
</workflow-forms>
```

Allow multiple choice

- Type is equal to **multiple**.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
    "http://www.openkm.com/dtd/workflow-forms-2.4.dtd">
<workflow-forms>
  <workflow-form task="select test">
    <select label="select label" name="select" type="multiple">
      <option label="one" value="001" />
      <option label="two" value="002" />
      <option label="three" value="003" />
    </select>
  </workflow-form>
</workflow-forms>
```

Options based on metadata table

- Application use two tables named `OKM_DB_METADATA_TYPE` AND `OKM_DB_METADATA_VALUE` that helps storing used options values. That's very useful when you have a large number of values.
- In the table `OKM_DB_METADATA_TYPE` are stored metadata table definition (table name, column name and column type).
- In the table `OKM_DB_METADATA_VALUE` are stored record values.

Create metadata table definition.



The **table** will be called **country**, with two columns, **country_id** and **country_name**.

The **country_id** column is **integer** type and **country_name** is **text** type.

The **country_id** is set as **column 0** with **col00** value.

The **country_name** is set as **column 1** with **col01** value.

- Go to **Administration > Database query**.
- At the right bottom select **JDBC** from list.
- **Paste the SQL** in the box

```
DELETE FROM OKM_DB_METADATA_TYPE WHERE DMT_TABLE='country';
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
```

- Click at bottom right the **Execute button**.

Insert metadata table values:

- Go to **Administration > Database query**.
- At the right bottom select **JDBC** from list.
- **Paste the SQL** in the box

```
DELETE FROM OKM_DB_METADATA_VALUE WHERE DMV_TABLE='country';
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('country',
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('country',
```

- Click the **Execute button** at the right bottom.



The insert queries depending on your database can not be matched, for example in Oracle:

```
INSERT INTO OKM_DB_METADATA_TYPE (DMT_ID, DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE,
DMT_VIRTUAL_COLUMN) VALUES (HIBERNATE_SEQUENCE.NEXTVAL, 'country', 'col00', 'integer',
'country_id');
```

More information at: [Creating your own database tables](#).

Finally the XML definition:

- In `optionsQuery` the values that begins with character \$ are identified as column names. See `$country_id` and `$country_name` below.
- Can use normal sql clauses like `"order by"` etc.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.6//EN"
    "http://www.openkm.com/dtd/workflow-forms-2.6.dtd">
<workflow-forms>
  <workflow-form task="select test">
    <select label="country" name="country" type="simple" table="country"
      optionsQuery="select $country_id, $country_name from DatabaseMetadataValue" />
  </workflow-form>
</workflow-forms>
```

Based on SQL query

We can also get these option values using a SQL query. This SQL sentence should be put on the `optionsQuery` attribute, but the `table` attribute should be empty (otherwise this query is used as a metadata query). Note that the first element returned in every row is the option value and the second one is the option label.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
    "http://www.openkm.com/dtd/workflow-forms-2.4.dtd">
<workflow-forms>
  <workflow-form task="select test">
    <select label="country" name="country" type="simple"
      optionsQuery="select CT_ID, CT_NAME from COUNTRY order by CT_NAME" />
  </workflow-form>
</workflow-forms>
```


The table used in this sample is defined as:

```
create table COUNTRY (
  CT_ID varchar(2),
  CT_NAME varchar(32),
  primary key(CT_ID)
);

insert into COUNTRY (CT_ID, CT_NAME) values ('es', 'Spain');
insert into COUNTRY (CT_ID, CT_NAME) values ('pt', 'Portugal');
insert into COUNTRY (CT_ID, CT_NAME) values ('it', 'Italy');
```

Workflow Separator field


Draw a separator between fields.

Attribute	Description	Required
label	The text shown as a label in user interface.	true
name	Unique field identifier  Two fields can't have the same name in the same task definion. Name must be unique.	true
width	The width of the HTML element.	false
readonly	Set field, it will not be able to be modified in user interface. By default the value is " false ".	false

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
    "http://www.openkm.com/dtd/workflow-forms-2.4.dtd">
<workflow-forms>
  <workflow-form task="separator test">
    <input label="Input label" name="input1" />
    <separator label="Separator label" name="separator" />
    <input label="Input label" name="input2" />
  </workflow-form>
</workflow-forms>
```

Workflow Suggestbox field

This is an implementation of the typical AJAX suggest box component:

Attribute	Description	Required
label	The text shown as a label in user interface.	true
name	Unique field identifier. <div>  Two fields can't have the same name in the same task definion. Name must be unique. </div>	true
table	Metadata table name. <div> Application can store values in a metadata table, that can be used in this query. </div>	true
width	The width of the HTML element.	false
height	The height of the HTML element.	false
filterQuery	A Hibernate query to filter the data. Use {0} to place the user input.	true
valueQuery	A Hibernate query to obtain the value of an identifier. Use {0} to place the identifier.	true
filterMinLen	The minimum chars in input to begin to filter results.	true
dialogTitle	The title of the HTML dialog.	true
readonly	Set field, it will not be able to be modified in user interface. By default the value is "false".	false
Child elements: <ul style="list-style-type: none"> Validator. For more information see Workflow Validator element. 		

Suggestbox values are defined in metadata tables.

- The application use two tables named `OKM_DB_METADATA_TYPE` AND `OKM_DB_METADATA_VALUE` that helps storing selected options values used. That's very useful when you get a large number of values.
- In the table `OKM_DB_METADATA_TYPE` are stored metadata table definition (table name, column name and column type).
- In the table `OKM_DB_METADATA_VALUE` are stored record values.

Create metadata table definition.



The **table** will be called **country**, with two columns, **country_id** and **country_name**.

The **country_id** column is **integer** type and **country_name** is **text** type.

The **country_id** is set as **column 0** with **col00** value.

The **country_name** is set as **column 1** with **col01** value.

- Go to **Administration > Database query**.
- At the right bottom select **jdbc** from list.
- **Paste the SQL** in the box

```
DELETE FROM OKM_DB_METADATA_TYPE WHERE DMT_TABLE='country';
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
VALUES ('country', 'col00', 'integer', null);
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
VALUES ('country', 'col01', 'text', null);
```

- Click at the right bottom of the **Execute button**.

Insert metadata table values:

- Go to **Administration > Database query**.
- At the right bottom select **jdbc** from list.
- **Paste the SQL** in the box

```
DELETE FROM OKM_DB_METADATA_VALUE WHERE DMV_TABLE='country';
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('country', '1', 'country');
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('country', '2', 'country');
```

- Click at the right bottom on the **Execute button**.



The insert queries, depending on your database can not be matched, for example in Oracle:

```
INSERT INTO OKM_DB_METADATA_TYPE (DMT_ID, DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE,
DMT_VIRTUAL_COLUMN) VALUES (HIBERNATE_SEQUENCE.NEXTVAL, 'country', 'col00', 'integer',
```

```
'country_id');
```


Finally the XML definition:

- In optionsQuery the values that begins with character \$ are identified as column names. See **\$country_id** and **\$country_name** below.
- Can use normal SQL clauses like "**lower**", "**like**", "**order by**", etc.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
    "http://www.openkm.com/dtd/workflow-forms-2.4.dtd">
<workflow-forms>
  <workflow-form task="suggestbox test">
    <suggestbox label="country" name="suggestbox" width="200px"
      table="country" dialogTitle="Choose country" filterMinLen="3"
      filterQuery="select $country_id, $country_name from DatabaseMetadataValue dmv where
        valueQuery="select $country_id, $country_name from DatabaseMetadataValue dmv where
    </workflow-form>
  </workflow-forms>
```

Workflow Text field


Print a text that can contain HTML tags.

Attribute	Description	Required
label	The text shown as a label in user interface.	true
name	Unique field identifier <div> Two fields can't have the same name in the same task definion. Name must be unique.</div>	true
width	The width of the HTML element.	false
height	The height of the HTML element.	false
data	When present, is an identifier used to load data.	false
readonly	Set field, it will not be able to be modified in user interface. By default the value is " false ".	false

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
    "http://www.openkm.com/dtd/workflow-forms-2.4.dtd">
<workflow-forms>
  <workflow-form task="text test">
    <text label="text label" name="consulting_text" width="200px" />
  </workflow-form>
</workflow-forms>
```

Workflow TextArea field


Print a text that can contain HTML tags.

Attribute	Description	Required
label	The text shown as a label in user interface.	true
name	Unique field identifier  Two fields can't have the same name in the same task definion. Name must be unique.	true
value	Default value for the component.	false
width	The width of the HTML element.	false
height	The height of the HTML element.	false
data	When present, is an identifier used to load data.	false
readonly	Set field, it will not be able to be modified in user interface. By default the value is " false ".	false
Child elements: <ul style="list-style-type: none"> Validator. For more information see: Workflow Validator element. 		

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
    "http://www.openkm.com/dtd/workflow-forms-2.4.dtd">
<workflow-forms>
  <workflow-form task="textarea test">
    <textarea label="textarea label" name="textarea" width="200px" />
  </workflow-form>
</workflow-forms>
```

Workflow Option element

The option element is only used in combination with [Workflow Select field](#). The option value represents a list of distinct values to select.

Attribute	Description	Required
label	The text shown as label in the user interface.	true
value	Unique value identifier. <div>  Two options can't have the same value. Value must be unique. </div>	true
selected	When present, it specifies that an option should be pre-selected when the form is rendered.	false
Parent : <ul style="list-style-type: none"> Select. For more information, Please see: Workflow Select field. 		

Basic select with selected option

- attribute selected=true, when present, it specifies that an option should be pre-selected when the page loads.


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.4//EN"
    "http://www.openkm.com/dtd/workflow-forms-2.4.dtd">
<workflow-forms>
  <workflow-form task="textarea test">
    <select label="select label" name="test" type="simple">
      <option label="one" value="001" />
      <option label="two" value="002" selected="true" />
      <option label="three" value="003" />
    </select>
  </workflow-form>
</workflow-forms>
```






Automation

Automation is a set of controls that perform actions based on a set of rules.

Preliminar concepts.

Each rule is composed of:

Field	Description
Order	<p>Indicates the order of the evaluation.</p> <div>  <p>The rules with less</p> </div>
Name	Rule description.
Events	<p>Available events:</p> <ul style="list-style-type: none"> • Document creation. • Document update. • Document move. • Folder creation. • Mail creation. • Adds metadata group. • Sets metadata group. • Text extraction. • Converts to PDF. • Converts to SWF. • User login. • User logout.
At	<p>The evaluation time.</p> <p>Available values:</p> <ul style="list-style-type: none"> • Pre. • Post.

	<p><i>Understanding stages on automation execution:</i></p> <div> <div>Pre</div> <div>Internal execution</div> <div>Post</div> </div> <div>  <p>Pre indicate rule execution before an internal event is executed.</p> <p>Post indicate rule execution after the internal event is executed.</p> </div> <div>  <p>For example a rules based on "Document creation" event.</p> <ul style="list-style-type: none"> On "pre" stage the rule is triggered when the document has not yet been created. On "post" stage the rules is triggered after the document has been created </div>
Exclusive	When enabled, indicates no order rules should be executed.
Active	Enables or disables the rule.
Validators	<p>List of validators. Each validator is a condition that must be complied.</p> <div>  <p>Validations are also ordered. It's a good practice to first set positions that will have more possibilities to comply.</p> </div> <div>  <p>When empty, always comply.</p> </div>
Actions	<p>List of actions. An action is something that will be done with the node that caused the execution of the rule.</p> <div>  <p>Nodes types:</p> <ul style="list-style-type: none"> Document Folder Mail <p>Example of an action, can be renaming a document immediately after creation.</p> </div>

Understanding Validators and Actions

By default application comes with built in validators and actions:

- [Built in Automation Validator plugins](#)
- [Built in Automation Action plugins](#)

Also you can create your own plugins:

- [Creating your own Automation Action](#)
- [Creating your own Automation Validation](#)

Creating a rule



Rules are composed of:

- General fields.
- Actions and validators definitions.

A rule creation has two steps. First define the general fields, then the actions and validations definitions.

- On top right click on **Create new rule.**
- **Fill in the fields.**
- Click on **Create button.** (the application will return to the initial screen).
- Click on **Definition icon.**
- Add actions and validations according to convenience.
 - Click on **Add validation icon.**
 - Click on **Add action icon.**
- When finished, at the top left click on **Automation rules button.** (the application will return to the initial screen).






Editing a rule

Edit general parameters

- Click on **Edit rule.**
- **Modify the fields.**
- Click on **Edit button.**

Edit action and validation definitions

- Click on **Definition icon.**
- Modify actions and validations according to convenience.
 - Validations
 - Click on **Add validation icon.**

- Click on  **Edit validation icon.**
- Click on  **Delete validation icon.**
- **Actions**
 - Click on  **Add action icon.**
 - Click on  **Edit action icon.**
 - Click on  **Delete action icon.**

Deleting a rule




- Click on  **Delete rule.**
- Click on **Delete button.**

Built in Automation Validator plugins

Name	Apply at	Parameters	Description	Rest
ContainsFolder	Post	Folder name	<p>Valid when the node path contains a folder name.</p> <div>  <p>The value of the parameter folder name is test. Node with path /okm:root/folder1 triggered the event.. the folder /okm:root/folder1/test exists ,then the validation return is valid.</p> </div> <div>  <p>The value of the parameter folder name is test. Node with path /okm:root/folder1 triggered the event.. A folder /okm:root/folder1/test doesn't exist, then the validation return is not valid.</p> </div>	Only appl to fo
HasCategory	Post			
HasKeyword	Post	Keyword name	Valid when the node path has a keyword with the same name assigned.	App. to al type.
HasPropertyGroup	Post	Metadata group name	Valid when the node path has a metadata group with the same name assigned.	App. to al type.
HasPropertyGroupValue	Post	Metadata field name Metadata field value	Valid when the node path has a metadata group field with the same value assigned.	App. to al type.
NameContains	Post	Name	Valid when the node name contains the name value into.	App. to al

			<div>  <p>The value of the parameter name is test.</p> <p>Node with path /okm:root/folder/sometest triggered the event.</p> <p>Folder sometest contains the value test ,then the validation return is valid.</p> </div> <div>  <p>The value of the parameter name is test.</p> <p>Node with path /okm:root/folder/somefolder triggered the event.</p> <p>Folder somefolder does not contains the value test ,then the validation return is not valid.</p> </div>	type.
PathContains	Post	Path	<p>Valid when the node path starts with a path.</p> <div>  <p>The value of the parameter path is /okm:root/folder1.</p> <p>Node with path /okm:root/folder1/test/temporary triggered the event.</p> <p>/okm:root/folder1/test/temporary starts with path /okm:root/folder1 ,then the validation return is valid.</p> </div> <div>  <p>The value of the parameter path is /okm:root/folder1.</p> <p>Node with path /okm:root/test/temporary triggered the event.</p> <p>/okm:root/test/temporary does not start with path /okm:root/folder1 ,then the validation return is not valid.</p> </div>	App. to al type.

Built in Automation Action plugins

<i>Name</i>	<i>Apply at</i>	<i>Parameter</i>	<i>Description</i>	<i>Restriction</i>
AddCategory	Post	Category	Adds a category to the node that triggered the event.	Applicable to all node types.
AddCategoryToWizard	Post		Shows a category popup as part of the process of uploading a new document. <div>  Only takes effect when an user uploads a new document from the user interface. </div>	Applicable to documents.
AddDigitalSignatureToWizard	Post		Shows a digital signature popup as part of the process of uploading a new document.	Applicable to documents.
AddKeyword	Post	Keyword name	Adds a keyword to the node that triggered an event.	Applicable to all node types.
AddKeywordTo Wizard	Post		Shows a keyword popup as part of the process of uploading a new document. <div>  Only takes effect when an user uploads a new document from user interface. </div>	Applicable to documents.
AddPropertyGroupToWizard	Post	Metadata group name	Shows a Metadata group popup as part of the process of uploading a new document. <div>  Only takes effect when an user uploads a new document from the user interface. </div>	Applicable to documents.

AddWorkflowToWizard	Post	Workflow name	<p>Starts a workflow as part of the process of uploading a new document.</p> <div>  <p>A popup will be shown if a workflow requires an user to sets parameters for starting the workflow. Otherwise, it will be transparent to the end user.</p> <p>Only takes effect when an user uploads a new document from user interface.</p> </div>	Applicable to documents.
ExecuteScripting	Pre Post	Code	Executes an script.	Applicable to all node types.
GrantRole	Post	Role name Permission	Grants a role permission the node that triggered the event.	Applicable to all node types.
GrantUser	Post	User name Permission	Grants a user permission the node that triggered the event.	Applicable to all node types.
OMRDataCapture	Post		Captures data from document.	Applicable to OMR valid files.
PurgeNode	Post		Purge a node. The node is removed from repository.	Applicable to all node types.
RevokeAllRoles	Post		Revokes all roles granted to the node that triggered the event.	Applicable to all node types.

				types.
RevokeAllUsers	Post		Revoke all user granted to the node that triggered the event.	Applicable to all node types.
RemoveCategory	Post	Category	Removes the category from the node that triggered the event.	Applicable to all node types.
RemoveKeyword	Post	Keyword name	Remove the keyword from the node that triggered the event.	Applicable to all node types.
RevokeRole	Post	Role name	Revokes the role granted to the node that triggered the event.	Applicable to all node types.
RevokeUser	Post	User name	Revokes the user granted to the node that triggered the event.	Applicable to all node types.

Enable Automation validations

Almost databases

- Go to **Administration** > **Database query**
- Choose **type JDBC**.
- Paste the SQL

```
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
```

- Click on  **Execute button**.

Oracle

- Go to **Administration** > **Database query**
- Choose **type JDBC**.
- Paste the SQL

```
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
```

- Click on  **Execute button**.

PostgreSQL

- Go to **Administration** > **Database query**
- Choose **type JDBC**.
- Paste the SQL

```
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
```

```
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, A
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, A
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, A
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, A
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, A
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, A
```

- Click on  **Execute button**.


```
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, 'A')
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, 'A')
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, 'A')
```

- Click on  **Execute button.**

PostgreSQL

- Go to **Administration > Database query**
- Choose **type JDBC**.
- Paste the SQL

[illegible]

- Click on  **Execute button.**

Crontab

The application implements a time-based job scheduler named Crontab. In crontab view there are a set a jobs that are executed periodically at fixed times or intervals.





Typically automated application maintenance or administration are crontab jobs.

Crontab is shown in a table structure.

Name	Expression	Mime	File name	Mail	Last begin	Last end	Active
------	------------	------	-----------	------	------------	----------	--------

Crontab parameters

Field	Description	Required
Name	Job name.	true
Expression	<p>Sets when the job is executed.</p> <div>  <p>Takes in consideration the status of the application when the task will be executed.</p> <p>For example, if you stop the application at 00:00 to perform a backup, it delays the task until you consider the backup process will be completed.</p> </div>	true
Mime	<p>Supported MIME types:</p> <ul style="list-style-type: none"> • application/x-bsh (BeanShell). • application/x-java-archive (JAR file). <div>  <p>The application supports tasks written in beanshell or classes into JAR file.</p> <p>We recommend using JAR files instead of beanshell.</p> <p>More information at: Creating your own Crontab job.</p> </div>	true
Mail	When present, a mail is sent as part of the task once the execution has been completed. You can use several mail addresses using a comma "," as separator.	false

<i>Last begin</i>	<i>Shows the last time a process started.</i>	-
<i>Last end</i>	<i>Shows the last time a process was completed.</i>	-
<i>Active</i>	<i>Enables or disables the process execution.</i>	<i>true</i>

Understanding expression syntax

The commands are executed by cron when the minute, hour, and month fields match the current time, and when at least one of the two day fields (day of month, or day of week) match the current time. The scheduler examines crontab entries once every minute. The time and date fields are:

```
* * * * command to execute
â", â", â", â", â",
â", â", â", â", â",
â", â", â", â", â""â"ëà"ëà"ëà"ë day of week (0 - 6) (0 to 6 are Sunday to Saturday,
â", â", â", â""â"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ë month (1 - 12)
â", â", â", â""â"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ë day of month (1 - 31)
â", â""â"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ë hour (0 - 23)
â""â"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ëà"ë min (0 -
```

<i>Field</i>	<i>Values</i>
<i>minute</i>	0-59
<i>hour</i>	0-23
<i>day of month</i>	1-31
<i>month</i>	1-12

Other expressions:

Expression	Equivalent	Description
@yearly	0 0 1 1 *	Runs once a year at midnight of January 1.
@annually	0 0 1 1 *	Same as @yearly.
@monthly	0 0 1 * *	Runs once a month at midnight of the first day of the month.

@weekly	0 0 * * 0	Runs once a week at midnight on Sunday morning.
@daily	0 0 * * *	Runs once a day at midnight.
@midnight	0 0 * * *	Same as daily.
@hourly	0 * * * *	Runs once an hour at the beginning of the hour.

A field may be an asterisk (*), which always stands for first-last.


Ranges of numbers are allowed. Ranges are two numbers separated with a hyphen. The specified range is inclusive. For example, "8-11" for an hours entry specifies execution at hours 8, 9, 10 and 11.

Lists are allowed. A list is a set of numbers (or ranges) separated by commas. Examples: "1,2,5,9", "0-4,8-12".


Step values can be used in conjunction with ranges. Following a range with "/<number>" specifies skips of the number's value through the range. For example, "0-23/2" can be used in the hours field to specify command execution every other hour. Steps are also permitted after an asterisk, so if you want to say "every two hours", just use "* /2".

Names can also be used for the month and day of week fields. Use the first three letters of the particular day or month (case doesn't matter). Ranges or lists of names are not allowed.


Create a new crontab job

- On top right click  **New crontab job icon**.
- **Fill the fields.**
- Click on **Create button**.


Edit a crontab job

- Click on  **Edit crontab job icon**.
- **Modify the fields.**
- Click on **Edit button**.


Execute a crontab job

- Click on  **Execute crontab job icon**.




Download a crontab job

- Click on  **Download crontab job icon.**
- After some second will be retrieved the file.

Delete a crontab job

- Click on  **Delete crontab job icon.**
- Click on **Delete button.**

Built in crontab tasks

Task	Default time	Execution	Description
Dashboard Activity Purge	@daily	At 00:00.	Clean older than one month activity log entries.
User Items Serialize	@hourly	Each hour.	
Refresh User Items	@weekly	Each sunday.	<p>Refresh the total repository size assigned to a single user.</p> <div>  Each user is owner of some repository size. </div>
User Mail Importer	* /30 * * * *	Each 30 minutes.	Import mails from registered user mail accounts.
Text Extractor Worker	* /5 * * * *	Each 5 minutes.	<p>Process text extraction queue.</p> <div>  For more information Statistics. </div>
Session Watchdog	* /5 * * * *	Each 5 minutes.	Kill orphan sessions.
Process Pending Tasks	* /5 * * * *	Each 5 minutes.	<p>Execute pending task.</p> <div>  For more information Statistics. </div>

OMR templates

Application integrates OMR - Optical Mark Recognition - that has been written by Aaditeshwar Seth from the Udai Waterloo chapter, at the University of Waterloo. Udai is a volunteer student organization at the University of Waterloo in Canada and the University of California at San Diego. For more information about it see [OMRProj website](#).

OMR (Optical Mark Recognition) engines is especially suited for extracting data from forms that have been photo-copied and then scanned; hence, the resultant images are likely to suffer from rotations, smudge marks, and random lines here and there.

Examples of a few forms that the tool is able to handle with 100% accuracy are shown here.

The image displays two examples of OMR forms. The left form is a detailed survey with the following sections:

- # of years in Thane**: 1, 2, 3, 4, 5+ (Options: 1, 2, 3, 4, 5+)
- # of years in this slum**: 1, 2, 3, 4, 5+ (Options: 1, 2, 3, 4, 5+)
- Name on electoral role**: Yes, No (Options: Yes, No)
- Ration card**: Yes, No (Options: Yes, No)
- Card No**: dd/mm/yy (Options: dd/mm/yy)
- Voter ID card**: Yes, No (Options: Yes, No)
- Mother tongue**: Marathi, Hindi, Kannada, Other (Options: Marathi, Hindi, Kannada, Other)
- Home**: Kaccha, Semi-pukka, Pakka, House (Options: Kaccha, Semi-pukka, Pakka, House)
- Vehicle**: None, 2-Whe, Bike, Other (Options: None, 2-Whe, Bike, Other)
- Electricity**: Own meter, Borrowed, No elec (Options: Own meter, Borrowed, No elec)
- Water**: Own, Public (Options: Own, Public)
- Toilet**: Own, Public, Other (Options: Own, Public, Other)
- Disposal**: Soak pit, Septic, Sewage (Options: Soak pit, Septic, Sewage)
- Garbage**: Bin, Open, Other (Options: Bin, Open, Other)
- Comforts**: Cable, Fan, Fridge, Newspaper, Tele, Mixer, TV (Options: Cable, Fan, Fridge, Newspaper, Tele, Mixer, TV)
- First name**: [Text field]
- Last name**: [Text field]
- Address**: [Text field]

The right form is a shorter version of a similar survey with the following sections:

- # of years in Thane**: 1, 2, 3, 4, 5+ (Options: 1, 2, 3, 4, 5+)
- # of years in this slum**: 1, 2, 3, 4, 5+ (Options: 1, 2, 3, 4, 5+)
- Name on electoral role**: Yes, No (Options: Yes, No)
- Ration card**: Yes, No (Options: Yes, No)
- Card No**: dd/mm/yy (Options: dd/mm/yy)
- Voter ID card**: Yes, No (Options: Yes, No)
- Mother tongue**: Marathi, Hindi, Kannada, Other (Options: Marathi, Hindi, Kannada, Other)
- Home**: Kaccha, Semi-pukka, Pakka, House (Options: Kaccha, Semi-pukka, Pakka, House)
- Vehicle**: None, 2-Whe, Bike, Other (Options: None, 2-Whe, Bike, Other)
- Electricity**: Own meter, Borrowed, No elec (Options: Own meter, Borrowed, No elec)
- Water**: Own, Public (Options: Own, Public)
- Comforts**: Cable, Fan, Fridge, Newspaper, Tele, Mixer, TV (Options: Cable, Fan, Fridge, Newspaper, Tele, Mixer, TV)
- First name**: [Text field]
- Last name**: [Text field]
- Address**: [Text field]

Add a new OMR template



Get a full example [Omr_example.zip](#)

Preliminars

- First make a form in PowerPoint using the template [Omr_patterns.zip](#). Make sure that you do not move the two concentric circles in the top-left and bottom-right corners of the template. These are used by the software to determine the angle of rotation of the form.
- Take a printout and make as many photocopies of the form as you want.
- On the original printout, color all the marks with a black pen, and scan this marked up printout. Please be as careful as you can to align the printout so that there is no rotation. Some amount of error is fine, but not too much! **This scanned image will be the template.**

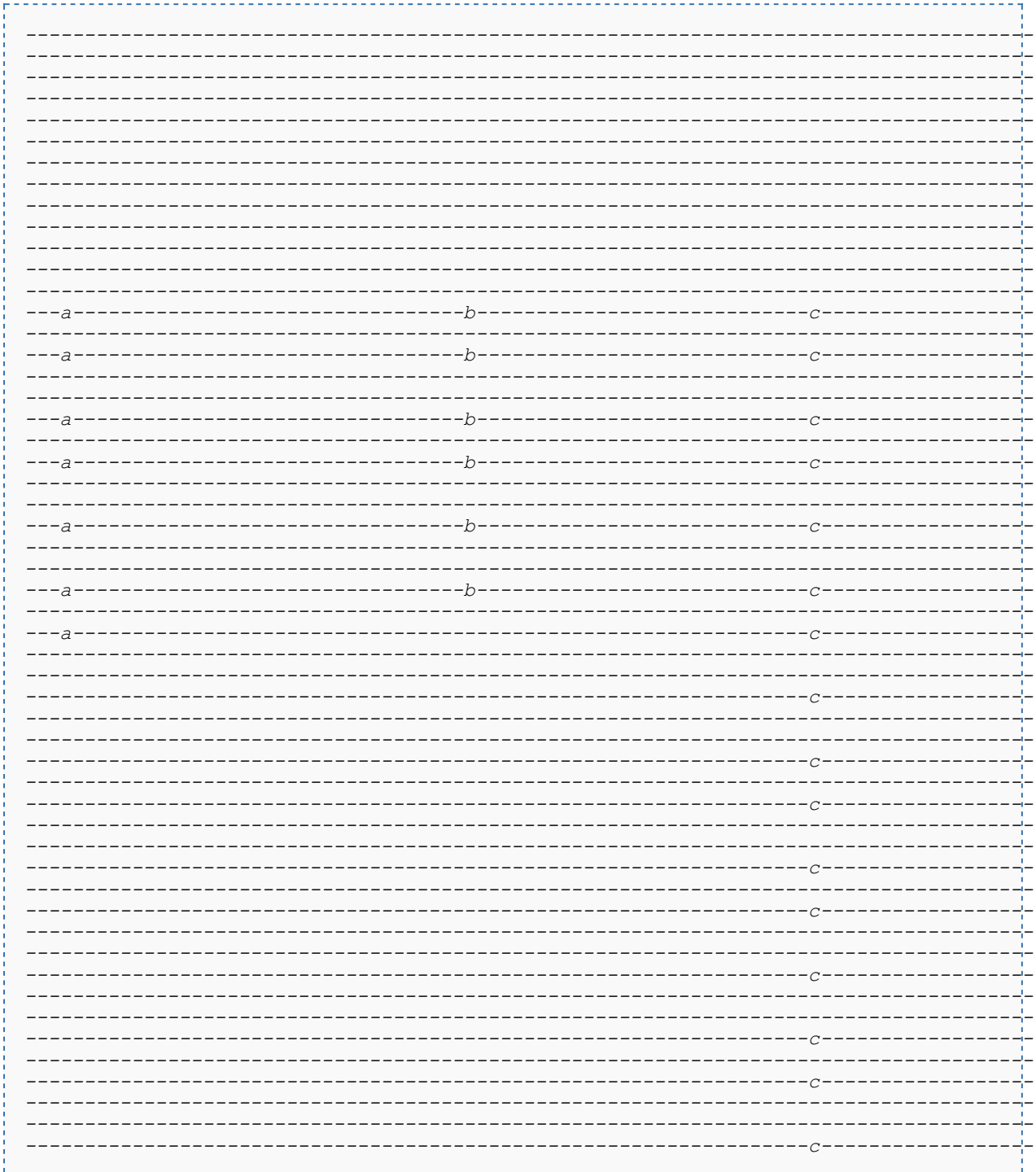


Seleccione Proyecto	Seleccione Año del Documento	Seleccione Tipo de Documento
<ul style="list-style-type: none"> ● Alarmas Calle Larga ● Alarmas Rinconada ● Alcantarillados Rinconada ● Mina Caracoles ● Colegio Metodista ● Obras Adicionales MC ● Villa Las Rosas 	<ul style="list-style-type: none"> ● 2012 ● 2013 ● 2014 ● 2015 ● 2016 ● 2017 	<ul style="list-style-type: none"> ● Actas ● Boleta de Compra ● Boleta de Honorario ● Boleta de Venta ● Comprobante de Compra ● Comprobante de Venta ● Contratos de Trabajo ● Contratos de Venta ● Documentación de Contabilidad ● Documentación de Propuesta ● Documentación Legal ● Factura de Compra ● Factura de Venta ● Manuales ● Oficios ● Otros Documentos






Some of the marks may not be exactly aligned, but don't worry. Now, very carefully, label the marks with single characters, by setting all the marks in the same group with the same character. This is shown below. In case you run out of alphabets, you can use numerals or any other character such as - + ? . etc. Make sure that you do not alter the position of any mark! This is agreeably hard to do, but we chose this method to avoid the need for making a graphical frontend.




The format of each line is: character type subtype name value1 value2....

Field	Description
Character	This is the same as the character you used in the .asc file
Type and Subtype	The types of fields supported are single and multiple choice lists, aligned row-wise (horizontally) or column-wise (vertically). And grids aligned horizontally or vertically. Each field is denoted by (i) row single or row multiple for a single or multiple selection field aligned row-wise, (ii) column single or column multiple for a single or multiple selection field aligned column-wise, or (iii) grid row or grid column for a grid field aligned row-wise or column-wise respectively.
Field name	This can be any name you want to associate with the field.
Values	<p>These are the field values, in order.</p> <div>  <p>The values has a direct corresponde with metadata field values, first indicate the field name and others the values into</p> </div>

Example:

```
a column single okp:data.project alarmasCalleLarga alarmasRinconada alcantarilladosRi.
b column single okp:data.year 2012 2013 2014 2015 2016 2017
c column single okp:data.type actas boletaCompra boletaHonorario boletaVenta comproba.
```

- Click on  **Edit template icon.**
- Click on  **Edit fields training file icon.**
- **Chose the training .fields file.**
- Click on **Send button.**

Omr template Edit omr template

Template name	<input type="text" value="Test template"/>
Template	template.png <input type="button" value="Examinar..."/>
Property	<div><div>Select properties ▾</div><div><div>okp:data.year ✕</div><div>okp:data.type ✕</div><div>okp:data.project ✕</div></div></div>
Asc	template.png.asc
Config	template.png.config
Fields	<div>Upload new file </div>
Active	<input checked="" type="checkbox"/>

Check the template

- Click on **Check icon**.
- **Choose some file to upload** (will be used to capture data).

Seleccione Proyecto	Seleccione Año del Documento	Seleccione Tipo de Documento
● Alarmas Calle Larga	● 2012	● Actas
● Alarmas Rinconada	● 2013	● Boleta de Compra
● Alcantarillados Rinconada	● 2014	● Boleta de Honorario
● Mina Caracoles	● 2015	● Boleta de Venta
● Colegio Metodista	● 2016	● Comprobante de Compra
● Obras Adicionales MC	● 2017	● Comprobante de Venta
● Villa Las Rosas		● Contratos de Trabajo
		● Contratos de Venta
		● Documentación de Contabilidad
		● Documentación de Propuesta
		● Documentación Legal
		● Factura de Compra
		● Factura de Venta
		● Manuales
		● Oficios
		● Otros Documentos

And you'll get the results:

Omr template
Process Template Check

Upload Form
Examinar...

Cancel
Send

RESULTS

Key	Value
okp:data.year	2014
okp:data.type	documentacionPropuesta
okp:data.project	minaCaracoles

Thesaurus


The thesaurus feature helps the administrator to create the thesaurus folder structure. By default the thesaurus folder structure is not created.



The application needs a configured parameter to create thesaurus folders, check [Thesaurus configuration](#).

Take a look at the full configuration example at [Creating your own thesaurus](#).

Build thesaurus

- Choose the level (the level indicates the tree depth that will be shown during creation process).
- On top right click on  Generate button.



A log will be displayed on the screen to monitor thesaurus folder creation. This process could take some hours depending on your thesaurus size.

Languages

The application can work with several languages, here are set the languages terms definitions.

Languages are shown in a table structure.

<i>Id</i>	<i>Flag</i>	<i>Name</i>	<i>Translation status</i>
------------------	--------------------	--------------------	----------------------------------

The "en-GB" translation is used by application as default translation.




Sometimes translations are not fully completed and some terms are missing. The "en-GB" is used to identify which are the missing terms and how many are.

When a term is missing in other language the "en-GB" term is used in its place.



en-GB translation can not be removed.


Create a new language

- Click on  **new language icon**.
- Fill the fields
 - The id must be compliant with ISO 639 (2 levels).
 - Language name
 - The icon should be 16x11 px
- Click on **Create button**.



More information about [ISO 639](#).

Edit a language

- Click on  **Edit language icon**.
- **Change the fields**.
- Click on **Edit button**.

Delete a language

- Click on  **Delete language icon**.
- **Change the fields**.
- Click on **Delete button**.

Edit language translations



On the right top the translation can be filtered by key.

- Click on  **Edit language translation icon**.

Languages translations are shown in a table structure.

Module	Key property	English translation	Translation
--------	--------------	---------------------	-------------




By default are **tree modules** that indicate where translations are used:

- frontend (used on frontend)
- mobile (used on mobile view)
- extension (used on frontend by extension modules)

Two key property can not be equals.

- **Change the fields.**
- Click on **Edit button**.

Export a language translations

- Click on  **Export language translations icon**.
- The application **retrieve a SQL file** with language translation into.

Import a language translations



Download your language package from [Language translations](#).

- On right bottom **choose a Language package file**.
- Click on **Add new translation button**.

Language translations

<i>Language</i>	<i>File</i>	<i>Number of terms</i>
Arabic	<i>OpenKM_6.3_ar-SD.sql</i>	653
Azeri	<i>OpenKM_6.3_az-AZ.sql</i>	1563
Belgian/Flemish	<i>OpenKM_6.3_nl-BE.sql</i>	824
Bosnian	<i>OpenKM_6.3_bs-BS.sql</i>	505
Bulgarian	<i>OpenKM_6.3_bg-BG.sql</i>	830
Català	<i>OpenKM_6.3_ca-ES.sql</i>	1169
Chinese simple	<i>OpenKM_6.3_zh-CN.sql</i>	1181
Chinese traditional	<i>OpenKM_6.3_zh-TW.sql</i>	1184
Czech	<i>OpenKM_6.3_cs-CZ.sql</i>	1107
Danish	<i>OpenKM_6.3_dk-DK.sql</i>	802
Deutsch	<i>OpenKM_6.3_de-DE.sql</i>	1536
Deutsch - Austria	<i>OpenKM_6.3_de-AT.sql</i>	823
Dutch	<i>OpenKM_6.3_nl-NL.sql</i>	1227
English	<i>OpenKM_6.3_en-GB.sql</i>	1605
English - USA	<i>OpenKM_6.3_en-US.sql</i>	506

<i>Español</i>	<i>OpenKM_6.3_es-ES.sql</i>	1605
<i>Español Colombia</i>	<i>OpenKM_6.3_co-ES.sql</i>	505
<i>Euskera</i>	<i>OpenKM_6.3_eu-ES.sql</i>	1563
<i>Farsi</i>	<i>OpenKM_6.3_fa-FA.sql</i>	505
<i>French</i>	<i>OpenKM_6.3_fr-FR.sql</i>	1535
<i>Gallego</i>	<i>OpenKM_6.3_gl-ES.sql</i>	505
<i>Greece</i>	<i>OpenKM_6.3_el-GR.sql</i>	1172
<i>Hungarian</i>	<i>OpenKM_6.3_hu-HU.sql</i>	506
<i>Indonesian</i>	<i>OpenKM_6.3_id-ID.sql</i>	506
<i>Italian</i>	<i>OpenKM_6.3_it-IT.sql</i>	1184
<i>Japanese</i>	<i>OpenKM_6.3_jp-JP.sql</i>	802
<i>Korean</i>	<i>OpenKM_6.3_ko-KR.sql</i>	1675
<i>Latvian</i>	<i>OpenKM_6.3_lv-LV.sql</i>	506
<i>Lithuanian</i>	<i>OpenKM_6.3_lt-LT.sql</i>	506
<i>Macedonian</i>	<i>OpenKM_6.3_mk-MK.sql</i>	506
<i>Norwegian</i>	<i>OpenKM_6.3_nb-NO.sql</i>	1175

Polish	<i>OpenKM_6.3_pl-PL.sql</i>	1229
Português do Brasil	<i>OpenKM_6.3_pt-BR.sql</i>	1328
Romanian	<i>OpenKM_6.3_ro-RO.sql</i>	506
Russian	<i>OpenKM_6.3_ru-RU.sql</i>	1250
Serbian	<i>OpenKM_6.3_sr-SR.sql</i>	853
Slovak	<i>OpenKM_6.3_sk-SK.sql</i>	1184
Suomi	<i>OpenKM_6.3_fi-FI.sql</i>	1060
Swedish	<i>OpenKM_6.3_sv-SE.sql</i>	506
Thai	<i>OpenKM_6.3_th-TH.sql</i>	506
Turkish	<i>OpenKM_6.3_tr-TR.sql</i>	1213
Vietnamese	<i>OpenKM_6.3_zh-VN.sql</i>	1135

How to create a new translation




Before starting with a new translation, please check if is available in [Language translations](#) section.

A sample of new language base on Oromo from Ethiopia


The first step is identifying what is your language code based on ISO 639 (2 level). Looking at the table of [language identifiers](#), the code might be "**om-ET**".

The second step will get the country flag, in this case, Ethiopia. Go to famfamfam.com website and download the flags icons. The flags icon name are based ISO 369 (one level), that means we are looking for a file named "**et.gif**".


The third step will be to create the new language. Go to Administration > [Languages](#) and create a new language with these values:

- Click on  **new language icon**.
- Fill the fields
 - The "id" might be "**om-ET**".
 - The "name" might be "Oromo".
 - The "icon" might be the file named "**et.gif**".
- Click on **Create button**.

Once the new language has been created must set the translations:

- Click on  **Edit language icon**.
- **Change the fields**.
- Click on **Edit button**.

Finally, think on sharing with the community your translation.

- Click on  **Export language translations icon**.
- The application **retrieves a SQL file** with language translation into.
- Contact us at the contact website form and we will tell you the way for sharing with us the translation.



Repository import

The repository importer helps administrator to import data from the file system.



Initial import (import standard folders and documents from your openkm server).

Import from previous openkm export.

Field	Description
Filesystem path	Server import path.
Repository path	<p>The repository destination path.</p> <p>Common values:</p> <ul style="list-style-type: none"> • /okm:root • /okm:templates • /okm:personal • /okm:mail
Metadata	When enabled, the importer assumes the metadata information (.okm files) are present.
History	<p>When enabled, the importer assumes all document versions of the document are present.</p> <p>More information about document versions format at Repository export.</p>
Restore UUIDs	<p>When enabled, the importer will keep the same node UUID.</p> <div>  <p>When application is integrated with thirdparty software, usually the thirdparty software stores in their own database UUID of the node as main reference. This references should not be lost, that's why on these scenarios are needed to keep out the UUID's.</p> </div>
<div>  <p>Specially when importing from previous openkm, export must take in consideration that you must use the same import options that had been chosen while exporting, otherwise the import will fails.</p> </div>	

Import

- *Set the file system path.*
- *Set the repository path.*
- *Check the repository import options.*
- *Click on **Import button**.*

Repository export

The repository export feature helps the administrator to export the whole or part of the OpenKM repository to the file system.

The output result will be the same folders and files structure that are set into the application.

Field	Description
Repository path	<p>The repository path to export.</p> <p>Common values:</p> <ul style="list-style-type: none">• /okm:root• /okm:templates• /okm:personal• /okm:mail
Filesystem path	<p>Server destination path.</p> <p>Linux example:</p> <div>/mnt/export</div> <p>Windows example:</p> <div>c:/export</div>
Metadata	When enabled, metadata values are exported.
History	When enabled, all documents versions are exported.

Export

- Set the repository path.
- Check the repository export options.
- Click on Export button.

Understanding output format

Case metadata and history are not checked

The result will be a folder structure with documents and mails into. Although a document has several versions only will be

exported the actual version.

Case only metadata is checked

The result will be a folder structure with documents and mails into. Although a document has several versions only will be exported the actual version.

For each folder, document or mail will be created an **.okm** file. For example for a file named "Administration.html" will be created in the same path a file named "Administration.html.okm". The **.okm** files contains into all the metadata information associated to the node in a JSON format.

Example of .okm file:

```
{ "uuid": "8299d059-7894-468b-ba48-3f3b59ebc37c",
  "author": "okmAdmin",
  "name": "redacted_481.pdf",
  "path": "/okm:root/redacted_481.pdf",
  "created": { "year": 2015,
    "month": 6,
    "dayOfMonth": 1,
    "hourOfDay": 17,
    "minute": 18,
    "second": 0 },
  "lastModified": { "year": 2015,
    "month": 6,
    "dayOfMonth": 2,
    "hourOfDay": 10,
    "minute": 47,
    "second": 39 },
  "language": "en",
  "keywords": [],
  "categories": [],
  "subscriptors": [],
  "notes": [ { "user": "system",
    "date": { "year": 2015,
      "month": 6,
      "dayOfMonth": 2,
      "hourOfDay": 10,
      "minute": 47,
      "second": 40 },
    "text": "New version 1.1 by okmAdmin: Signed",
    "path": "7d11d393-df46-49a1-8f29-ae6d0ea4a093" } ],
  "propertyGroups": [],
  "version": { "author": "okmAdmin",
    "name": "1.1",
    "created": { "year": 2015,
      "month": 6,
      "dayOfMonth": 2,
      "hourOfDay": 10,
      "minute": 47,
      "second": 39 },
    "size": 139215,
    "mimeType": "application/pdf",
    "comment": "Signed" },
  "grantedUsers": { "okmAdmin": 15 },
  "grantedRoles": { "ROLE_USER": 15 }
}
```



Take in consideration that the JSON output format can change if the main objects (folder, document, mail or record) are added, removed or modified fields.

Case only history is checked

The result will be a folder structure with documents and mails. All document versions will be saved.

Example of output for a document with version 1.0 and 1.1:

- Administration.html (size = 0 bytes)
- Administration.html#v1.0# (contents of version 1.0)
- Administration.html#v1.1# (contents of version 1.1)

Case metadata and history are checked

The result will be a folder structure with documents and mails. All document versions will be saved.

For each folder, document or mail will be created a **.okm** file. For example for a file named "Administration.html" will be created in the same path a file named "Administration.html.okm". The **.okm** file contains all the metadata information associated to the node in a JSON format.

Example of output for a document with version 1.0 and 1.1:

- Administration.html (size = 0 bytes)
- Administration.html.okm (metadata associated to the document)
- Administration.html#v1.0# (contents of version 1.0)
- Administration.html#v1.0#.okm (metadata associated to version 1.0)
- Administration.html#v1.1# (contents of version 1.1)
- Administration.html#v1.1#.okm (metadata associated to version 1.1)

Utilities

The utilities section at Administration is a collection of useful features designed to be used by IT staff.

To take advantage of most of these feature some technical skills are needed .

These features can be categorized as:

Category	Description
Security	<ul style="list-style-type: none">• Purge permissions
Performace	<ul style="list-style-type: none">• Cache stats• Hibernate stats• Profiling stats
Low level access	<ul style="list-style-type: none">• Repository view
Check specific configuration	<ul style="list-style-type: none">• Check email• Check text extraction• System properties
Search engine	<ul style="list-style-type: none">• List indexes• Rebuild indexes
Others	<ul style="list-style-type: none">• Repository checker
Experimental features	<ul style="list-style-type: none">• LogCat• CSS

Cache stats

OpenKM takes advantage of cached objects for fast retrieving data.



For example the users and roles list, when application is configured to working with LDAP, are cached to reduce the traffic between application and LDAP server.

The associated cached name for LDAP are:

- `com.openkm.cache.ldapPrincipalAdapter.general`
- `com.openkm.cache.ldapPrincipalAdapter.name`


Also cache can be used to collect data useful to find out configuration parameters changes needed for better application performance. For this purpose the application comes with a cache audit trail.

Cache stats are shown in a table structure:

Cache name	Hits	Misses	Objects	Memory hits	Memory objects	Disc hits	Disc missed	Disc objects	Action
Field	Description		Audit trail required						
Cache name	Unique class name of the cached class.		Not applicable.						
Hits	Number of successful queries.		true						
Misses	Number of failed queries.		true						
Objects	Number of cached objects.		false						
Memory hits	Number of successful queries in cache memory.		true						
Memory missed	Number of failed queries in cache memory.		true						
Memory objects	Number of objects in cache memory.		false						
Disc hits	Number of successful queries in cache disk.		true						

Disc missed	Number of failed queries in cache disk.	true
Disc objects	Number of objects in cache disk.	false
Action	Clean cached objects.	Not applicable.

Clean all cache

- On the right top click on  **Reset all button.**

Refresh cache stats

- On the top right click on  **Refresh icon.**

Enable cache stats audit trail

- On the top right click on  **Enable cache audit trial button.**

Enables the fields stats:

- Hits.
- Misses.
- Memory Hits.
- Memory missed.
- Memory objects.
- Disc hits.
- Disc missed.

Disable cache stats audit trail

- On top right click on  **Disable cache audit trial button.**

Disable the fields stats:

- Hits.
- Misses.
- Memory Hits.
- Memory missed.
- Memory objects.
- Disc hits.

- *Disc missed.*

Clean cache stats audit trail

- On top right click on  ***Clear cache audit trail button.***

Reset to zero the value of the fields stats:

- *Hits.*
- *Misses.*
- *Memory Hits.*
- *Memory missed.*
- *Memory objects.*
- *Disc hits.*
- *Disc missed.*

Check email

The check email feature helps the administrator to check the application mail configuration.

Check email

- Fill the fields
 - From
 - To
 - Subject
 - Content
- Click on **Check button**



If the application mail configuration application is correctly set it will return an "OK" otherwise will be shown the cause of the error.

Check text extraction

This feature helps the administrator to check the text extraction process.

It can check the extraction process based on:

- *Document to be processed*
 - *Document UUID (see tab properties when document is selected on general application user interface).*
 - *Document path.*
 - *Document outside application (choose a document from file system to be uploaded).*

Check text extraction

- **Choose a document** to be processed
 - OpenKM document:
 - Fill field document UUID or document path.
 - External document:
 - Choose a document from file system to be uploaded.
- Click on **Check button**.



After the process is completed it will show the text extracted by the selected text extractor.



The text extraction process can take some seconds or minutes depending on the size of the document to be processed and the file MIME type.

Hibernate stats

With the Hibernate stats feature the administrator can take control of value information about database queries, finding out performance problems and the possible causes.



By default Hibernate stats feature is disabled because the log analysis consumes hardware resources.

This feature should only be enabled by administrator in order to investigate possible performance problems with application.



Each time the application is restarted all the previous analysis is removed. It's important that when the administrator enables this feature, takes in consideration that the application should not be stopped.

Information provided by Hibernate stats

Information provided by Hibernate stats are shown in several tables:

- General data.
- Query statistics.
- Entity statistics.
- Collection statistics.
- 2nd level cache statistics.



Time columns are finished with dur. name:

- Max dur (Maximum duration).
- Min dur (Minimum duration).
- Avr dur (Average duration).
- Total dur (Accumulated total time).

Columns with time values are in milliseconds.




When the application presents some performance problem, in most cases it will be caused by some query that need a huge time to complete its execution.

On Query statistics are shown, all the database queries executed by application, the most important column value is **Max dur**, which indicates the Maximum duration of the query.


Enable Hibernate stats



This option is only available when Hibernate stats is stopped.

- On right top click on?  Enable button.


Clear Hibernate stats

- On right top click on?  Clear button.

Disable Hibernate stats



This option is only available when Hibernate stats is running.

- On right top click on?  Disable button.


List indexes

The List index feature helps the administrator to take control about how the objects (documents, folders, mails and records) are indexed into Lucene search engine.



At the main screen is shown a table with the Lucene indexed fields of each object.

Common indexed fields:

Field	Description
#	Number
_hibernate_class	<p>The hibernate class name:</p> <ul style="list-style-type: none"> • <i>com.openkm.dao.bean.NodeFolder</i> • <i>com.openkm.dao.bean.NodeDocument</i> • <i>com.openkm.dao.bean.NodeMail</i>
uuid	Object unique identifier.
parent	Parent node.
context	<p>Context values:</p> <ul style="list-style-type: none"> • <i>okm_root</i> • <i>okm_metadata</i> • <i>okm_categories</i> • <i>okm_thesaurus</i> • <i>okm_templates</i> • <i>okm_personal</i> • <i>okm_mail</i> • <i>okm_personal</i> • <i>okm_trash</i>
tenant	Tenant identifier.

author	<i>Id of the user author.</i>
created	<i>Indicate when object was created.</i>
name	<i>Object name.</i>
userPermission	<i>User permissions.</i>
rolePermission	<i>Role permissions.</i>
 Depending the type of object (<code>_hibernate_class</code>) can be more indexed fields.	

Search indexes

- On top right click on  **Search indexes button.**
- In input box copy some object UUID (from desktop view, you can get at Properties tab any object UUID, select the UUID and press CTRL+C to copy).
- Click on **Search button.**
- From the table list click on  **List icon.**



The filtering input box use the sintaxis **FIELD : value** , here some examples:

- `uuid:ce3338a5-b1ec-481b-9d59-f5aeb3fcf42b`
- `name:invoice.*`
- `text:telephone`

Let's see an example. Imagine you have two documents called "Parana.doc" and "Parana.pdf". You can locate them by this search:

```
name:paran*
```

But be careful because using quotation marks you won't get any result:

```
name:"paran*"
```

If you want to locate all PDF documents which name starts with "Paran", the right query is:

```
name:paran*pdf
```

This query would also work (note the dot before the extension):

```
name:paran*.pdf
```

*But this query will fail because you cannot use a * or ? symbol as the first character of a search. This is a Lucene limitation.*

```
name:*pdf
```

If you want to lookup documents with a given extension, I would recommend using the MIME Type property this way:

```
mimeType:application/pdf
```

In this query we are looking for all documents which name starts with "parana" and has a keyword called "dog":

```
name:parana* AND keyword:song
```

You can learn more about query syntax at [Lucene Query Syntax](#).

Enable show terms

When show terms enabled, is shown an extra field named terms on documents objects with the terms used by Lucene search engine for indexing that document.

- On top right click on  **Show terms button**.

Disable show terms

- On top right click on  **Show terms button**.

LogCat

The LogCat feature helps administrator to accessing Tomcat log files directly from application.

Logs are shown is a table structure:

File	Action
------	--------




The main Tomcat log file is named **catalina.log**.


This file rotate each day at night creating a new catalina.log file and saving older with name catalina.log.year-month-day

Purge all logs


The purge option will clean all Tomcat logs.

- On top right click on  **Purge button**.


Download file as ZIP


- Choose a log file.
- Click on  **download as ZIP icon**.

Browser log file

- Choose a log file.
- Click on  **browse icon**.
- Use filter options.

Filter options:

Option	Description
Begin	<p>Indicate the row position.</p> <div>  By default 0 indicate the starting of the document. </div>
End	<p>Indicate the row position.</p> <div> By default -1 indicate the end of the document. </div>

	
String	<i>Filter by some text.</i>

Profiling stats

The Profiling stats feature gives low level value information to the administrator to find out application performance problems.



By default the Profiling stats feature is disabled because the log analysis consumes hardware resources.

This feature should only be enabled by the administrator in order to investigate possible performance problems with application.




Each time the application is restarted all the previous analysis are removed. It's important when administrator enable this feature take into consideration that the application should not be stopped.

Profiling stats are shown is a table structure:


Class	Method	Calls	Max dur.	Min dur.	Avg dur.	Total dur.	Action
Field	Description						
Class	The Java class name.						
Method	The Java method name into the class.						
Calls	Number of calls.						
Max dur.	Maximum duration of the calls.						
Min dur.	Minimum duration of the calls.						
Avr dur.	Average duration of the calls.						
Total dur.	Accumulated duration time.						
Action	List of all calls.						

The duration time is in milliseconds.


Enable profiling stats

- On top right click on  **Enable button**.

Disable profiling stats

- On top right click on  **Disable button**.


Clear profiling stats

- On top right click on  **Clear button**.

Refresh profiling stats

- On top right click on  **Refresh button**.

List of detailed calls

- Click on  **List detailed calls icon**.

Purge permissions

The purge permissions feature helps administrators to delete users and roles grants assigned to objects (documents, folders, mails or records).




Examples of usage:

- Remove grants of users or roles that no longer exists into the repository.
- Remove grants of users or roles that has been propagated along the repository accidentally.

Purge



This operation can not be reverted, use with care.

- From checkbox **choose users or roles** option.
- Click on **List Button**.
- Click on  **Delete icon**.

Rebuild indexes

The *Rebuild Indexes* feature helps the administrator to rebuild Lucene indexes.

<i>Indexes</i>	<i>Description</i>
<i>Text extractor</i>	<i>Extracts all document contents and rebuild the indexes.</i>
<i>Lucene indexes</i>	<i>Rebuild indexes with actual contents.</i>
<i>Optimize indexes</i>	<i>Remove older contents that no longer exists in repository.</i>

Rebuild

- Choose the index to rebuild.
- Click on ***Execute button***.

Repository checker

The repository checker feature helps the administrator to verify the repository integrity.



When the administrator thinks the system could be compromised, for example a hard disk failure or similar, he can use Repository checker feature to verify the application integrity.

Option	Description
Path	<p>The path from where will be started the verification process.</p> <p>Common values:</p> <ul style="list-style-type: none">• /okm:root• /okm:trash• /okm:personal• /okm:mail• /okm:templates
Fast	The fast verification will only check the existence of the last document version.
Versions	Verify the existence of all document versions. This option has priority over fast option.
Checksum	Calculate the checksum of repository document and compares with database value.

Check repository



The process can take several minutes or hours depending on the repository size and the repository options checked.

The repository checker process will show a process status in %, before it is shown the application calculates the total amount of nodes that should be checked in a huge repository this calculation process can take several minutes, please be patient.

- Set the path.
- Choose repository checker options (fast, versions, checksums).
- Click on **Check button**.

Repository view

The repository view feature provides a low level access to the repository that should be used with care because it allows to apply changes into the repository directly.

Node folder selected

Sections	Description
Info	<p>Get general information of the node selected.</p> <p>Allowed options:</p> <ul style="list-style-type: none"> • Path helps to navigate to parent nodes. • Remove contents action will remove all children nodes. • Remove current action will remove selected node and descendants.
Properties	<p>Shows the node properties:</p> <ul style="list-style-type: none"> • UUID. • Keywords. • Categories.
Property group	Shows and modifies the assigned metadata values.
Children	Shows all the children nodes (document, folders, mail and records into).

Node document selected

Sections	Description
Info	<p>Gets general information of the node selected.</p> <p>Allowed options:</p> <ul style="list-style-type: none"> • Path helps to navigate to parent nodes. • Remove current action will remove selected node. • Force text extraction action will execute the text extraction process.

Properties	Shows the node properties: <ul style="list-style-type: none"> • <i>UUID</i>. • <i>Keywords</i>. • <i>Categories</i>.
Property group	Shows and modifies the assigned metadata values.
History	Shows and downloads any version of the document. Shows the document differences between previous document versions.
Text extracted	Shows the text extracted from the document.

Node mail selected

Sections	Description
Info	Gets general information of the node selected. Allowed options: <ul style="list-style-type: none"> • Path helps to navigate to parent nodes. • Remove current action will remove selected node and descendants.
Properties	Shows the node properties: <ul style="list-style-type: none"> • <i>UUID</i>. • <i>Keywords</i>. • <i>Categories</i>.
Property group	Shows and modifies the assigned metadata values.
Children	Shows all the children nodes (documents into).

System properties

The system properties feature show the JVM variables.

Most common JVM variables:

Key	Description of Associated Value
<code>java.version</code>	Java Runtime Environment version
<code>java.vendor</code>	Java Runtime Environment vendor
<code>java.vendor.url</code>	Java vendor URL
<code>java.home</code>	Java installation directory
<code>java.vm.specification.version</code>	Java Virtual Machine specification version
<code>java.vm.specification.vendor</code>	Java Virtual Machine specification vendor
<code>java.vm.specification.name</code>	Java Virtual Machine specification name
<code>java.vm.version</code>	Java Virtual Machine implementation version
<code>java.vm.vendor</code>	Java Virtual Machine implementation vendor
<code>java.vm.name</code>	Java Virtual Machine implementation name
<code>java.specification.version</code>	Java Runtime Environment specification version
<code>java.specification.vendor</code>	Java Runtime Environment specification vendor
<code>java.specification.name</code>	Java Runtime Environment specification name
<code>java.class.version</code>	Java class format version number
<code>java.class.path</code>	Java class path
<code>java.library.path</code>	List of paths to search when loading libraries
<code>java.io.tmpdir</code>	Default temp file path
<code>java.compiler</code>	Name of JIT compiler to use

<code>java.ext.dirs</code>	<i>Path of extension directory or directories</i>
<code>os.name</code>	<i>Operating system name</i>
<code>os.arch</code>	<i>Operating system architecture</i>
<code>os.version</code>	<i>Operating system version</i>
<code>file.separator</code>	<i>File separator ("/" on UNIX)</i>
<code>path.separator</code>	<i>Path separator (":" on UNIX)</i>
<code>line.separator</code>	<i>Line separator ("\n" on UNIX)</i>
<code>user.name</code>	<i>User's account name</i>
<code>user.home</code>	<i>User's home directory</i>
<code>user.dir</code>	<i>User's current working directory</i>

CSS

The application allows to set your own CSS - Cascade Style Sheets - to be applied in user interface.



The CSS applied to user interface is set on [Profile General tab](#) section.



That helps the administrator to change the look and feel of the application.

For example, change the font size for cyrillic or logographic languages.

Css are shown in a table structure.

Name	Context	Active	Actions
------	---------	--------	---------

Create a new CSS



Can not be two css with same name.

- At the top right click on **New CSS icon**.
- **Fill in the fields.**
 - Name.
 - Context available are frontend or mobile.
 - Active.
- Click on **Create button**.

Edit a CSS

- Click on **Edit CSS icon**.
- **Modify the fields.**
- Click on **Edit button**.

Delete a CSS


- Click on **Delete CSS icon**.
- Click on **Delete button**.

Clone a CSS

- Click on **Clone CSS icon**.

- **Modify the fields.**
- Click on **Clone button**.

Download a CSS

- Click on  **Download CSS icon**.
- **Elapsed some seconds, will be retrieved a CSS file.**

Samples

Large font sample

```
body, td, a, p, div, input, input.submit , textarea, select, button{
  font-family: Tahoma, Arial, Helvetica, sans-serif;
  font-size: 13px;
}

.gwt-MenuItem, .okm-TopMenuBar .gwt-MenuItem-selected, .okm-Input, .okm-TextArea, .okm-Image-Popup .Caption {
  font-family: Tahoma, Arial, Helvetica, sans-serif;
  font-size: 13px;
}

.okm-Popup .Caption, .okm-Popup-Error .Caption, .okm-Image-Popup .Caption {
  font-family: Tahoma, Arial, Helvetica, sans-serif;
  font-size: 15px;
}
```

Medium font sample

```
body, td, a, p, div, input, input.submit , textarea, select, button{
  font-family: Arial, Helvetica, sans-serif;
  font-size: 12px;
}

.gwt-MenuItem, .okm-TopMenuBar .gwt-MenuItem-selected, .okm-Input, .okm-TextArea, .okm-Image-Popup .Caption {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 12px;
}

.okm-Popup .Caption, .okm-Popup-Error .Caption, .okm-Image-Popup .Caption {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 14px;
}
```

Default font sample

```
body, td, a, p, div, input, input.submit , textarea, select, button{
  font-family: Arial, Helvetica, sans-serif;
  font-size: 11px;
}

.gwt-MenuItem, .okm-TopMenuBar .gwt-MenuItem-selected, .okm-Input, .okm-TextArea, .okm-Image-Popup .Caption {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 11px;
}
```

```
.okm-Popup .Caption, .okm-Popup-Error .Caption, .okm-Image-Popup .Caption {  
  font-family: Arial, Helvetica, sans-serif;  
  font-size: 13px;  
}
```

User guide



You can use the [Quick start](#) guide to quickly learn the basics, and to find detailed info on the user sections.

OpenKM frontend is the Back office view on the content repository. The features are organized in several sections:

- The **Menu bar** at the top where are available all the features organized by type.
- The **Quick search box** at the right top.
- The **Toolbar** where are the most commonly used features.
- The **Desktop** tab enables to navigate the repository.
- The **Search** tab enables to query the repository.
- The **Dashboard** tab provides a set of dashboards.
- The **Administration** tab provides a set of the administration tools.
- The **Status** panel at the bottom provides general information.



Only members with "**ROLE_ADMIN**" role have access to the **Administration** tab.

The OpenKM frontend allows you to manage any type of content: scanned images, photographs, office documents and even large videos files or CAD files among others. The built-in workflow feature helps companies to automate businesses processes and get a full control of the cycle of the documents, saving time and money. It also allows the social activities around the content to be used to connect people to other people, information to information, and people to information; helping to manage, more efficiently, the collective intelligence of the human resources of the company.

The OpenKM [Profiles](#) feature helps administrators to set what is displayed for each user of the application.

OpenKM provides the solution to:

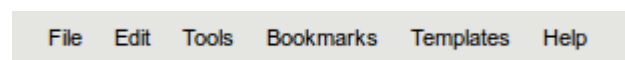
- Controls the growing volume of information to be managed.
- Ends the diversification of information storage sources.
- Prevents the duplication of the information.
- Obtains valid information through the control of document versions.
- Reduces the time and resources used in locating information.
- Avoids knowledge leaks or wrongful access by users, therefore complying with the obligations set in the LOPD (Spanish Data Protection Law).
- Restricts non-authorized users access to the information.

Through the OpenKM system, all relevant and useful information can be accessed and found in the less possible time to resolve problems and make decisions.

Menu bar

The menu bar at the top has all features organized by type:

- File.
- Edit.
- Tools.
- Bookmarks.
- Templates.
- Helps.



Quick search box

The Quick search box is at the top right corner. It provides a fast access to Default search in Search panel.



The minimal text length for doing a query is three characters.

ToolBar

The toolbar section has the most commonly used features like download documents, create new folders, edit or upload files among others.



Desktop

Most user's activity is done there. Users can work directly on folders, documents, mails or records.

The desktop is distributed in tree sections:

- **Navigation** panel at left hand side.
- **Browser** panel at top center.
- **Properties** panel at bottom center.

Navigation

The navigation at left hand side is an stack panel. Each stack contains into a tree (folder and records hierarchy) structure

from root that enables easily access documents and mails.

Browser

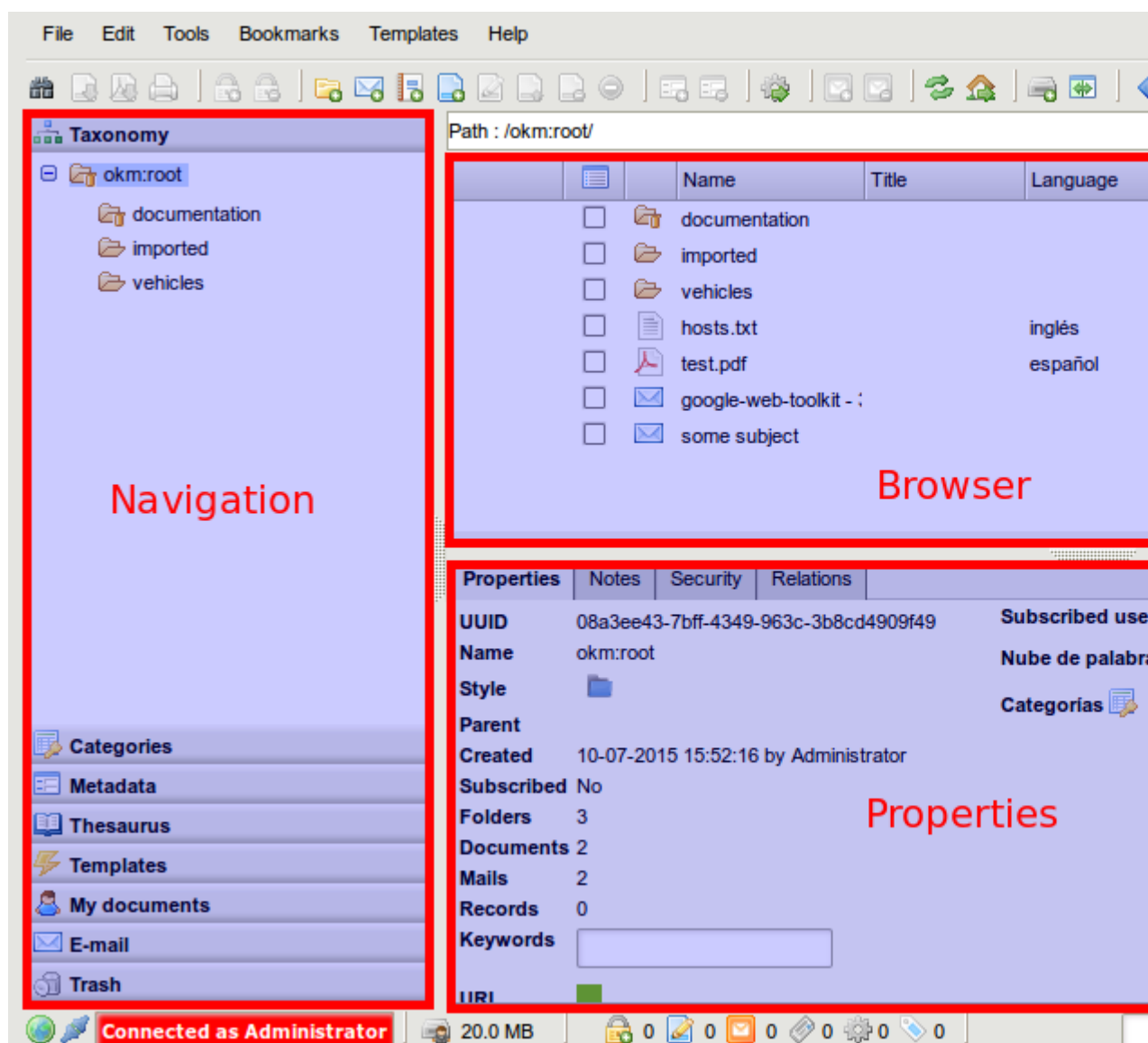
The browser panel at top center shows a table of nodes (documents, folders, mails and records) that are on the selected node of navigation panel.

Properties

The properties panel at bottom center shows a collection of tabs that contain the grouped properties or features like preview associated with selected node. The selected node can be a navigator node or browser node.



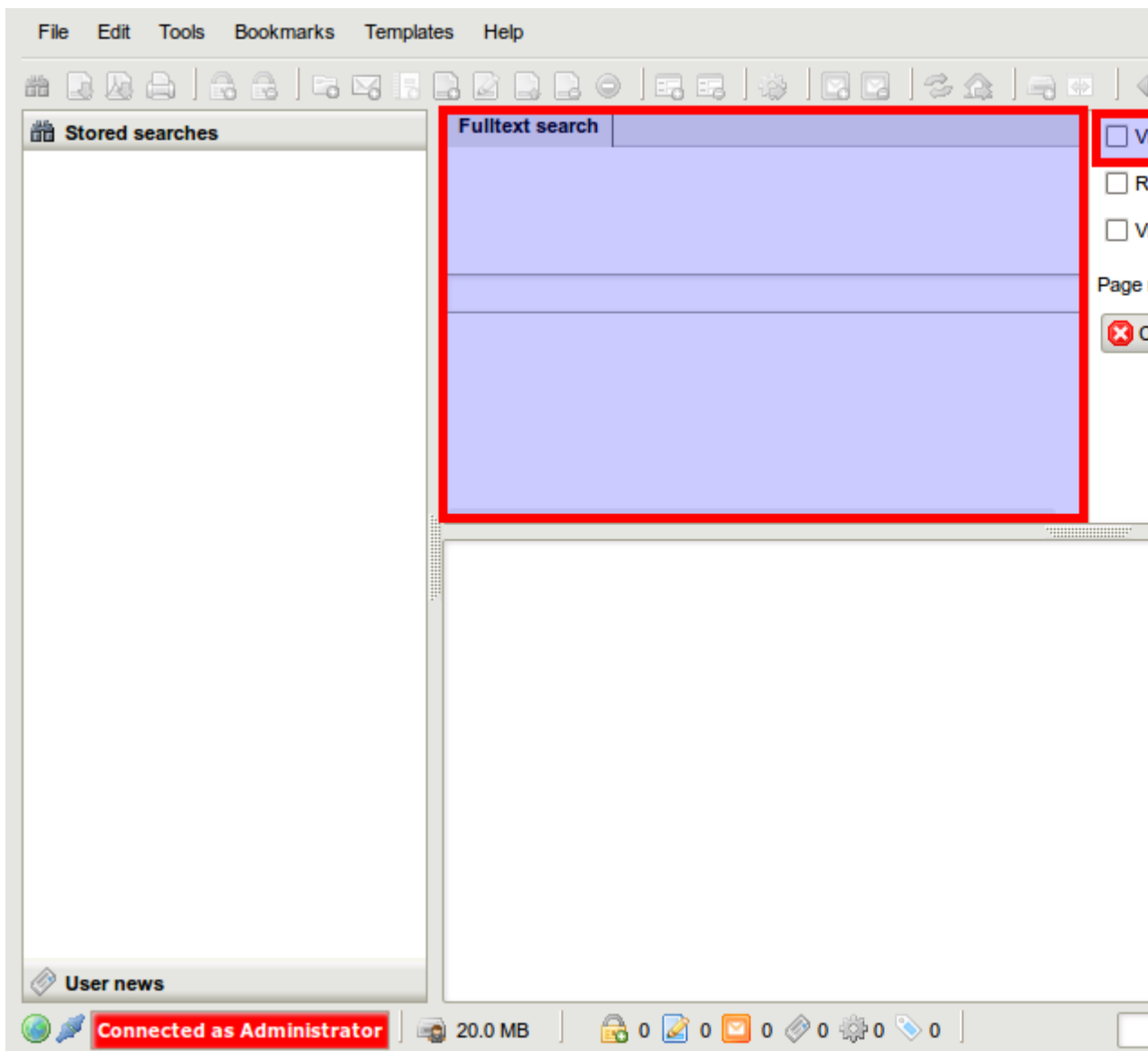
Basically the selected node at navigator panel determines the content shown at the browser panel and depending if the last selected node by the user is at navigator or browser panel the properties panel will shown the properties of the active node at navigator or browser.



Search

The search tab enables to query the repository. This tab gives access to different searches to query the content of the application:

- **Default search** is a full-text search. Default search is also available from the top right corner on the screen.
- **Advanced search** enables you to search keywords, metadata and additional criteria such as location in the application.



Dashboard

The dashboard panel provides a set of dashboards, each user has a set of personal dashboard that display the information relevant for him and helps to have a global view of the application activity.

The dashboard panel :

- **User view.** Displays the actions that have been carried out in the system
- **General view.** Displays a global view of the application activity.
- **News.** Displays information based on scheduled user search queries.
- **Mail.** It's a virtual navigation across all mail repositories.
- **Workflow.** Users work in company business process tasks.
- **Keyword map.** Displays a tag cloud based on all repository keywords.

- **Messaging.** Displays internal OpenKM messaging.
- **Forum.** Displays a forum.
- **Wiki.** Displays a wiki.
- **Meeting.** Manages meetings with OpenMeetings.

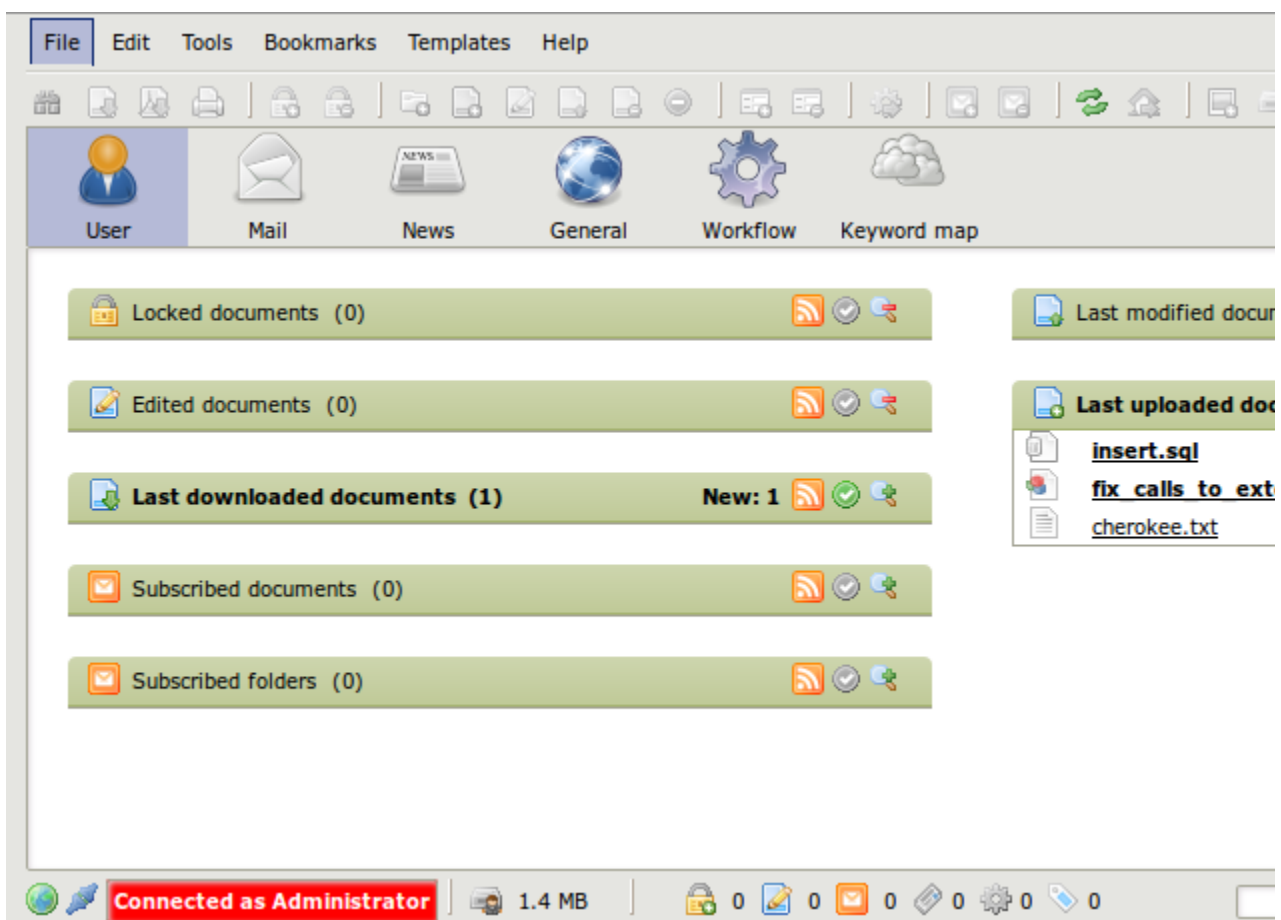


Messaging is only available if you have enabled the **Messaging** feature.

Forum is only available if you have enabled the **Forum** feature.

Wiki is only available if you have enabled the **Wiki** feature.

Meeting is only available if you have enabled the **OpenMeetings** feature.



Administration

The administration panel is a set of tools that helps administrator to manage the application.



More information at [Administration guide](#).

Status

The status panel provides general information:

- *OpenKM news alert.*
- *The connected user.*
- *User repository size.*
- *Chat.*
- *Alerts.*
- *HTML documents on edition.*
- *Message panel.*



Quick start



Before starting a production environment you should also read these sections:

- [Essential fundamentals](#).
- [Plan your data organization](#).

With OpenKM it's simply to store and share information and collaborate effectively.

This quick start guide give you an introduction to some of the features of the OpenKM:

- Logging in.
- Create a taxonomy.
- Add documents.
- Edit documents.
- Search documents.

Loggin in

You need to log in to OpenKM.

- Enter the following on the login page:
 - **Username:** your username.
 - **Password:** your password.
- Choose your **language**.
- Click on **Login** button.



Ask your administrator for your login URL. By default from the OpenKM server the URL is <http://127.0.0.1:8080/OpenKM>



The default administrator credentials are:

- **Username:** okmAdmin
- **Password:** admin

Do not forget to **change the administrator password**.

Create a taxonomy



The dictionary says:

"Taxonomy (from Greek taxis meaning arrangement or division and nomos

meaning law) is the science of classification according to a pre-determined system, with the resulting catalog used to provide a conceptual framework for discussion, analysis, or information retrieval. In theory, the development of a good taxonomy takes into account the importance of separating elements of a group (taxon) into subgroups (taxa) that are mutually exclusive, unambiguous, and taken together, include all possibilities."


*Basically, a **taxonomy** is a way to group things together. Like a controlled vocabulary focuses on concepts, in practice, a good taxonomy should be simple, easy to remember, and easy to use.*

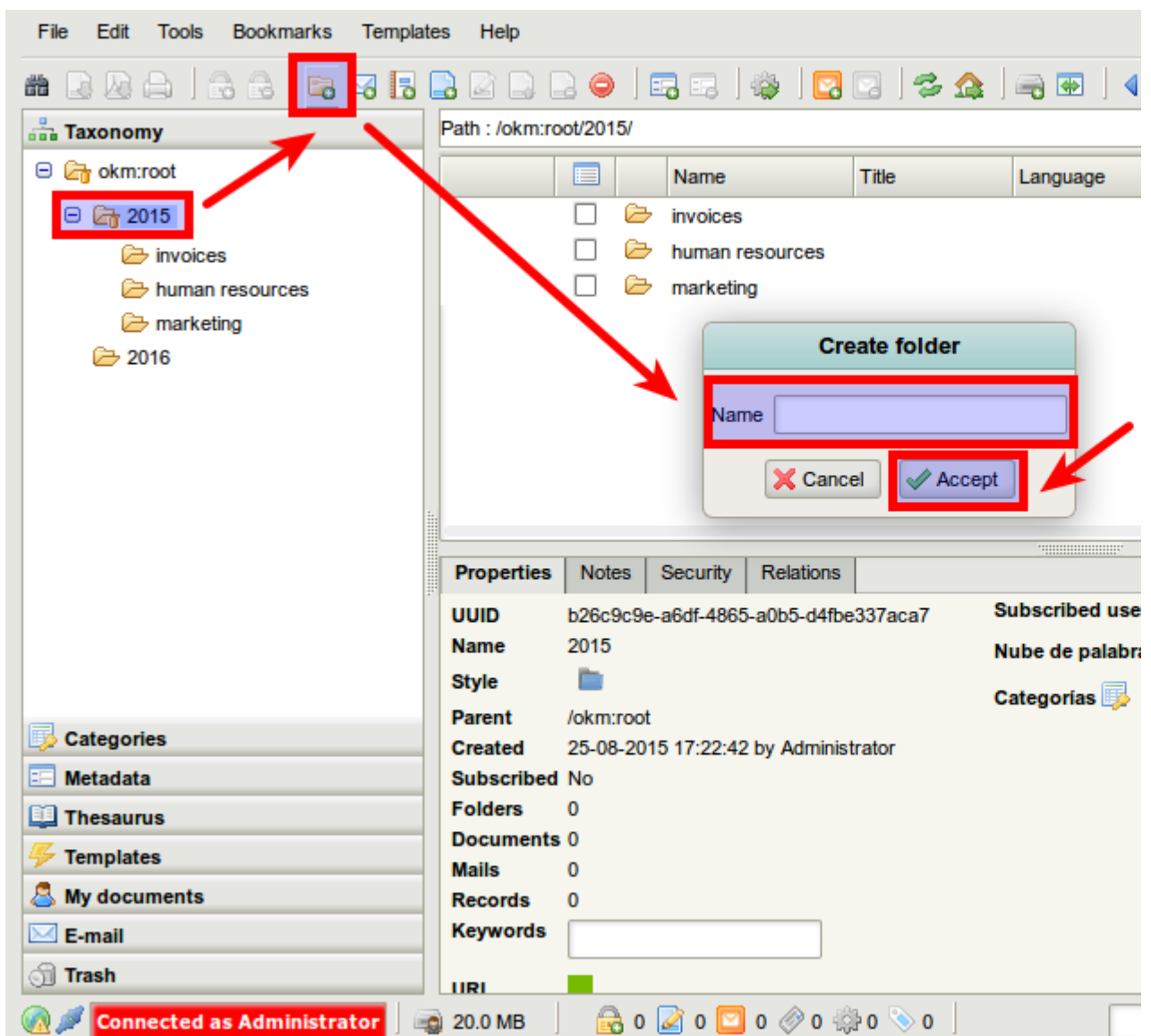
The first step should be analyse the company organization and define a set of folder hierarchy to store the documents organized.

Take this one as a sample:

/okm:root/	2015/	invoices/
		human resources/
		marketing/
	2016/	invoices/
		human resources/
		marketing/

To create a folder:

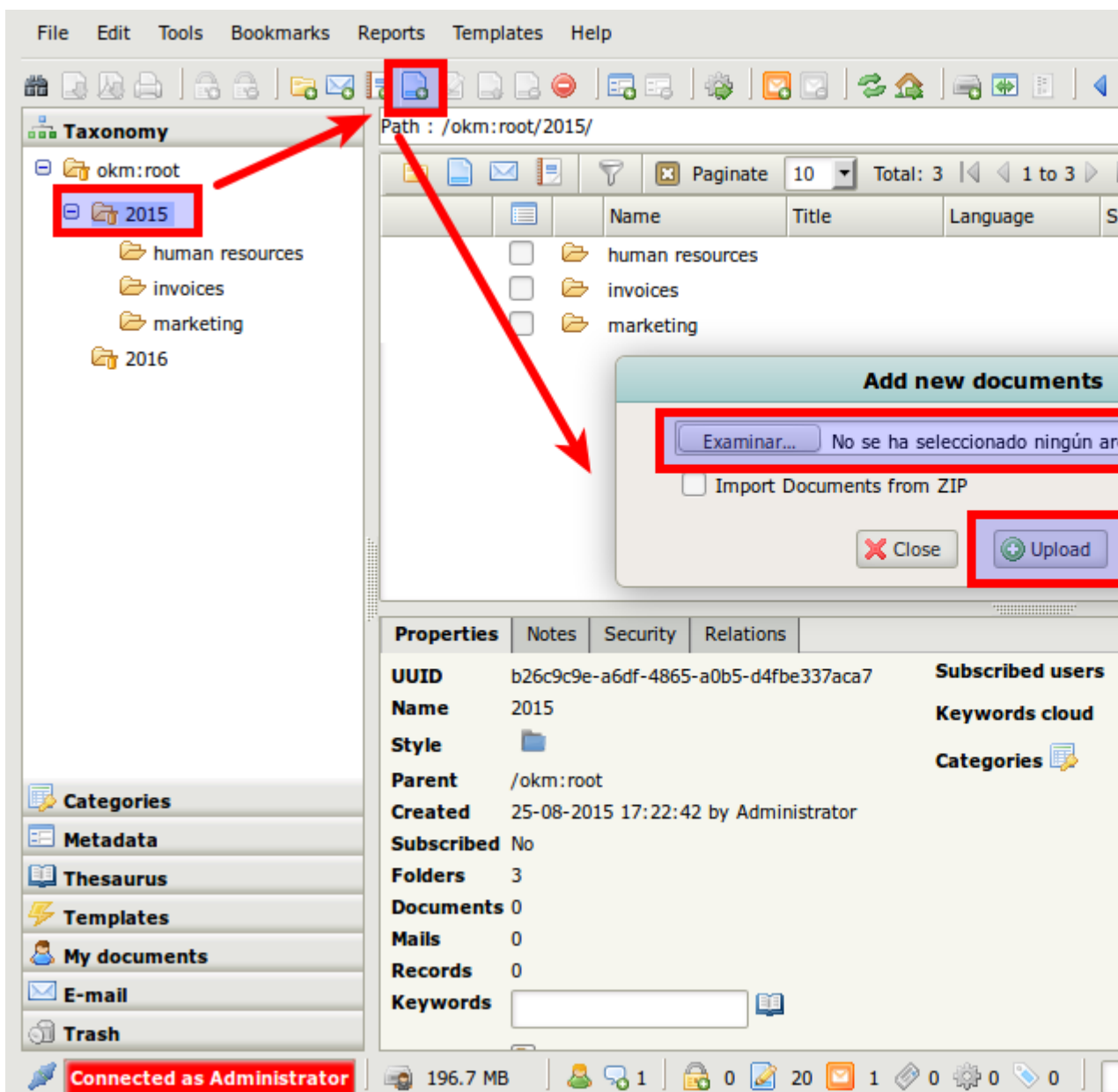
- **Select a node** from taxonomy where to create the folder.
- Click on  **folder icon** at tool bar.
- Set the **name**.
- Click on **Accept** button.



We suggest spending some time thinking and designing your taxonomy. For a best UI performance we suggest less than 200 elements per folder.

Add documents

- Select a **node** from taxonomy where to add the document.
- Click the **add document** icon from the toolbar.
- Select the **document** from your computer.
- Click **Upload** button.



Edit documents

The document edition is based on the "**check-out** / **check-in**" model.



"**Check-out**" and "**check-in**" are common terms in many document management systems, including OpenKM. Their purpose is to prevent conflicts in an environment where multiple people might want to edit the same piece of content - in OpenKM a document - at the same time.


The term "**check-out**" describes the process of getting a version of a document. By checking out a document, a user can prevent others from editing that content.

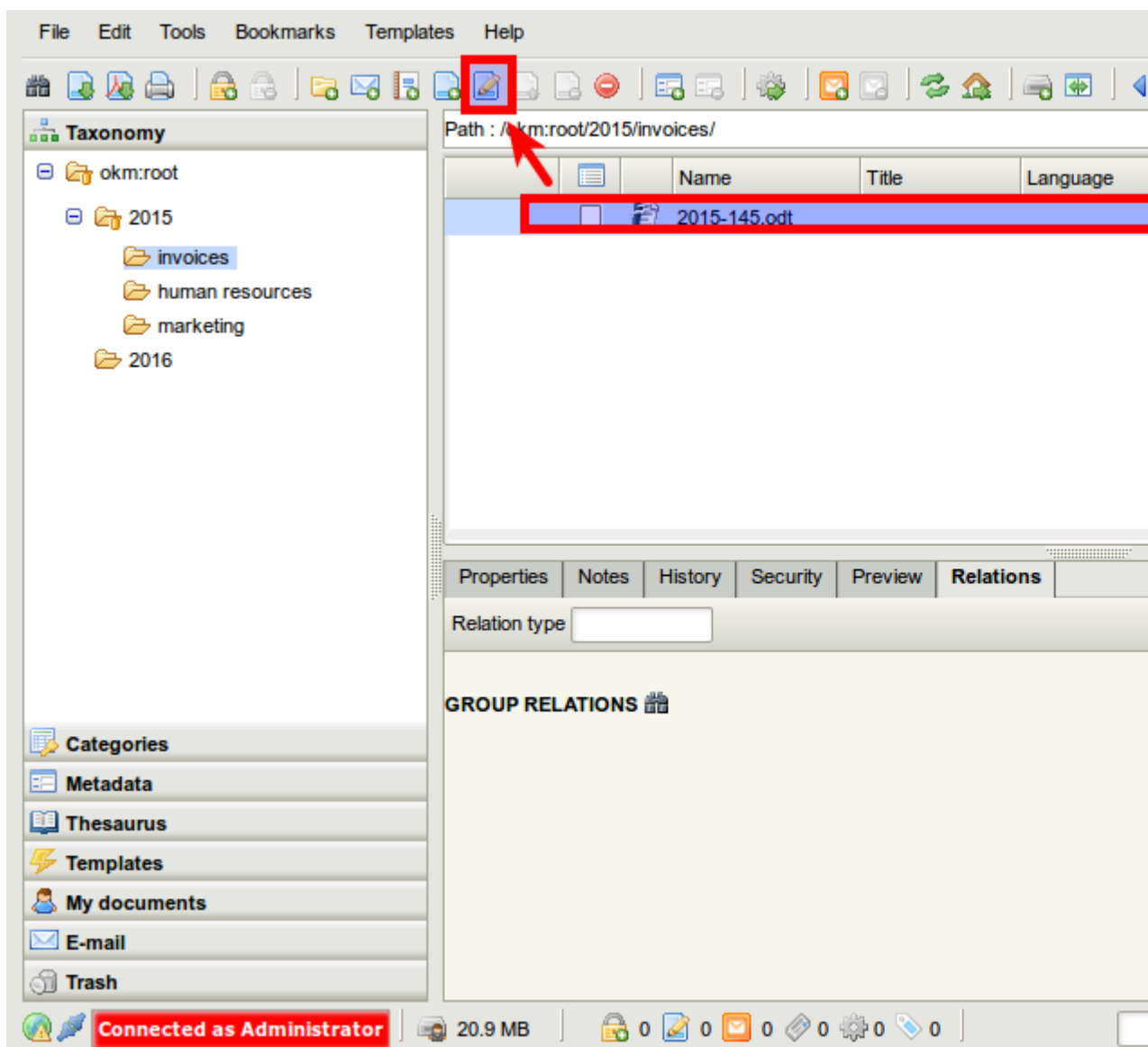
The term "**check-in**" describes the process of updating a document to replace the previous version. By checking in the item, the user can allow others to edit the document, without needing to worry about overriding changes.

that others have made.

When a document is checked out to a user, that user can work on that document, and other users cannot. Other users must wait for the user to check in the document before they can edit. While the document is checked out, other users usually can see only the last version that was checked in; they can't see the changes that the current user has made on the document while it's been checked out.


First must check-out the document:

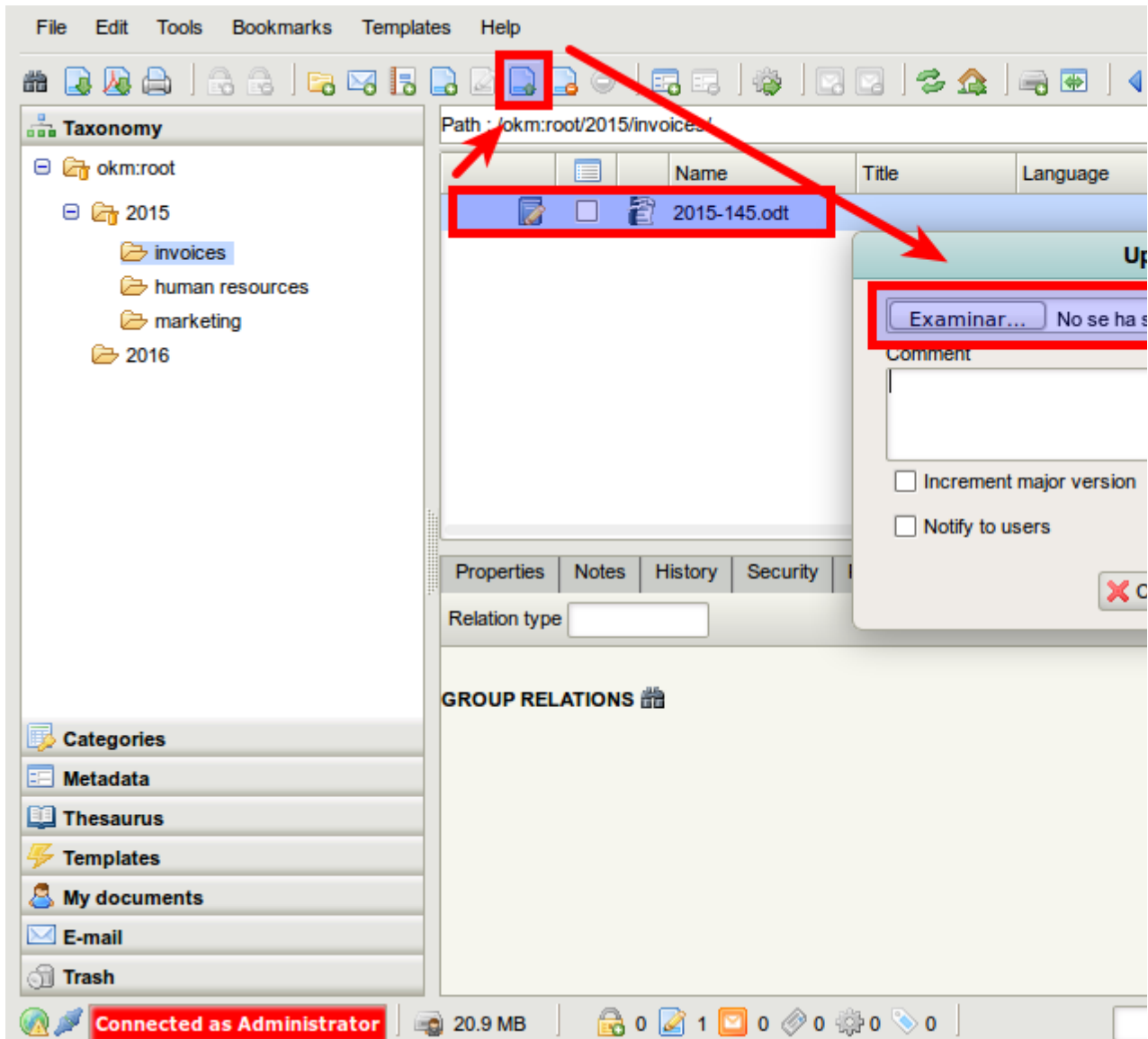
- **Select a document.**
- Click on  **Edit icon.**



Finally check-in the document to upload the changes:

- Select the **edited document**.

- Click on  **Update icon**.
- **Select the document** from your computer.
- Click **Upload** button.




Search documents

- Go to **search view**.
- **Write the term** for the query.
- Click on **Search** button.

The screenshot displays the OpenKM 6.3 CE web interface. The top menu bar includes File, Edit, Tools, Bookmarks, Templates, and Help. Below the menu is a toolbar with various icons. The main interface is divided into two main sections: a left sidebar and a right content area.

The left sidebar contains a 'Stored searches' section and a 'User news' section at the bottom. The right content area features a 'Fulltext search' section with a search input field containing the text 'system'. To the right of the search input, there are checkboxes for 'V' and 'R', and a 'Pa' button. Below the search input, a red box highlights the search results table.

Relevance	Name	Title	Size	Update
★★★★★	 2015-145.odt		29.4 KB	25-

At the bottom of the interface, a status bar shows 'Connected as Administrator' in a red box, followed by '20.9 MB' and several icons with counts (0).

Essential fundamentals

Basic concepts

OpenKM has built-in with several types of nodes:

- Folder.
- Document.
- Mail.

A node type **folder** is a **storage unit or a container**. As a document, it also has a name but is not a file, but it can contain documents and other folders into.

A node type **document** is a **file**. A document serves to establish one or several facts, and can be relied upon as a proof thereof.

A node type **mail**, most commonly referred to as **email** or **e-mail**, is a method of exchanging digital messages from an author to one or more recipients. A mail can contain documents (in mail context are named attachments).

Understanding taxonomy, keywords, categories and metadata groups.

Spending time on nodes categorization is a good practice. A good categorization will help users quickly finding fast of the desired information, using less time, saving money and being more efficient in your business process.

OpenKM helps you on it providing basically four ways for recategorizing your repository:

- Taxonomy.
- Keywords.
- Categories.
- Metadata groups.



You can use one of these features to categorize your repository or a combination of all.

Taxonomy

The dictionary says:

"Taxonomy (from Greek taxis meaning arrangement or division and nomos meaning law) is the science of classification according to a pre-determined system, with the resulting catalogue used to provide a conceptual framework for discussion, analysis, or information retrieval. In theory, the development of a good taxonomy takes into account the importance of separating elements of a group (taxon) into

subgroups (taxa) that are mutually exclusive, unambiguous, and taken together, include all possibilities."

Basically, a **taxonomy** is a way to group things together. Like a controlled vocabulary focuses on concepts, in practice, a good taxonomy should be simple, easy to remember, and easy to use.



For a best performance at UI we suggest - when possible - a taxonomy where each parent has 200 children or less.

Each folder in a taxonomy represents a "concept" and lists a couple of nodes (folders, documents or mails) associated within.

Keywords

A keyword, in the context of document management is a particular word or phrase that describes the contents. Setting keywords is the most simple and basic system to categorizing the nodes in your repository but at same time is the less powerfull. For large repositories we encourage using metadata groups.

Limitations of keywords usage:

- When you have thousands of keywords searching by keywords can be inefficient (not able to refine search and you get a lot of results with same keywords, the users not followed a restricted way to catalog). A tag cloud diagram in this cases is usually an unusable graphical representation of keywords.
- It's difficult for the users setting keywords because they do not know all the keyword dictionary of the company.
- Users write keywords in similar way but not always the same (for example users use the words "building", "construction" and "edifice" indistinctly).



You can get best results when:

- You limit the number of users who are able to set keywords.
- You use a controlled keywords dictionary (more information at <http://www.w3.org/2004/02/skos/>).
- You share a complete company keywords dictionary to your users. That helps understanding how is mapped the knowledge in your organization and helps users finding it.

Categories

In OpenKM, the categories are a virtual taxonomy that helps on navigation of taxonomy nodes across other concepts. Each category node represents a "concept" and list a couple of nodes (folders, documents or mails) associated within.



We suggest using categories rather than keywords.

Metadata

Metadata groups are data that describes other data providing information about a certain nodes content. Metadata groups helps on creating controlled classification and a powerful way to locate nodes and refine query searches.



With metadata groups you can simulate the keywords and categories features. We encourage expend some time investigating metadata feature and we suggest use it rather keywords and categories.

Plan your data organization

Before and intensive use of your application you should plan your data organization in the repository.



Take the information that follows only as a general idea about the basic steps you should consider before an intensive use of your application.

A good analysis of your organization and the understanding of the application can help you finding the best way to solve your problems. OpenKM is not a closed application, something more than a couple of features focused on document management. OpenKM is a flexible application and can be used on several ways. It's good idea to expend some time understanding your organization, your business logic and process involved and how the OpenKM features can work together in your day-by-day helping your organization making it more efficient and saving money.

Some questions you should do yourself about your organization:

- Where is your information and documents ?
- What kind of documents do you have ?
- Are your documents on electronic format or not ?
- What format of document do you have ? PDF, MS Office, OpenOffice, etc.
- How are you actually finding the document ? How would you like to find it, which terms you consider relevant ?
- Who can access the information and on which way ?
- Do you need a work flow process ?

Which are the objective of the previous questions ?

- Identify on a table all the distinct types of information you have (invoices, budgets, legal papers, etc.).
- Identify the format of the documents (PDF, DOC, etc.)
- Identify non digitalized information that should be processed (scanned). The digitalization process can be a basic process.
- Identify for each type of information (document type) what information is relevant (metadata). For example invoices usually have invoice number, range of date, customer among others. The metadata will help users to retrieve fast and efficiently the information.
- Identify the security that must be applied for each document type. Who can read, write or delete information. Also your security needs will help you on setting the level of the application log you need. OpenKM provide you a full audit on any user action that can be configured in a more or less restrictive way.
- Identify workflows and people involved in workflows.



The goal is to create a plan with calendar and milestones.

Which are the goals you are looking for the implantation of the application ? Usually could be:

- *Set some order in my organization. Application "force" the users to work in some way, that benefits your company because helps you "setting rules" that all users must follow.*
- *Consolidate the intellectual capital. Consolidating all your company knowledge on a single point you have all the company knowledge available to be share among all your users and setting rules on it.*
- *Help on accomplish with legal regulation.*



Try to set and specification on a single document answering this questions and others. It will help you to plan how to organize the information into.

Good practice

- *Consider OpenKM as a project that affects all your departments and usually your customers.*
- *Choose the leaders.*
- *Do gradual implantation of the project. Can start with some department or group of users or for some business logic process. Think the implantation as a virus in your company. A good leader should demonstrate the benefits of the new and show to the other users. Users usually receipt any news on the company as something negative if they perceive the application as something is helping on their work (efficient search, saving time and solving real problems, etc.) will be the first interested on using it. One of the major goals of OpenKM is the intuitive UI what helps on this process (after two weeks the user feels comfortable using the application).*
- *Think easy, the best solution is usually finding the easiest way to solve problems.*
- *Listen your users and try to get them involved in the project.*
- *Periodically review your plan, try to refine and identify possible problems.*

Working in menu bar

At the menu bar on the top all available features are organized by type. The menu options is formed by all the pull down menus with their different options. If an option within a menu appears greyed-out, this option is not available

The menu bar options are:

- File.
- Edit.
- Tools.
- Bookmarks.
- Templates.
- Help.

















The features described below are the default features. In your menu bar options can be shown others not described here if you have OpenKM extensions enabled.
















Massive column in the tables, indicates you can check several nodes at the same time in browser panel to apply the same action.

File

Feature	Description	Massive
 Find folder	<p>Finds folder in the repository.</p> <div>  The search can be executed only on the context of the selected stack. </div>	No.
 Find document	<p>Finds a document in the repository.</p> <div>  The search can be executed only on the context of the selected stack. </div>	No.
 Find similar document	<p>Finds similar documents in the repository. The feature is based on the search engine heuristic that is based on contents of the selected document.</p> <div>  The search can be executed only on the context of the selected stack. This feature is only enabled when a document in Browser view is selected . </div>	No.

 Create folder	<p>Creates a new folder into the selected node of the navigator panel.</p>	No.
 Add document	<p>Uploads a new document into the selected node of the navigator panel.</p> <div data-bbox="347 533 1281 925" style="border: 1px dashed #add8e6; padding: 10px; margin: 10px 0;"> <p>i When the option "Import documents from ZIP" is checked you import the contents of a ZIP file respecting the hierarchy of the folder tree. If you have problems with filename encodings, please take a look at Client configuration . You can also upload RAR or JAR MIME-TYPE files.</p> <p>When the option "Notify to users" is checked you can send a mail notification to other application users.</p> <p>Optionally, based on Administration > Profiles configuration or Administration > Automation a wizard is shown. Wizard can require to you for setting metadata groups, categories or keywords.</p> </div>	No.
 Download document	<p>Downloads the selected document in browser panel.</p>	No.
 Download document as PDF	<p>Download the selected document in browser panel in PDF format.</p> <div data-bbox="347 1272 1281 1361" style="border: 1px dashed #ffcc00; padding: 10px; margin: 10px 0;"> <p> The feature is enabled when is possible convert the document to PDF format.</p> </div>	No.
 Send document attachment	<p>Sends the selected document in browser panel as an attachment by mail.</p>	Yes.
 Forward e-mail	<p>Forwards the selected mail in a browser panel.</p>	No.
 Create from	<p>Creates a new document based on an existing one at templates space.</p>	No.

template		
 Export to ZIP	Downloads the selected nodes compress into ZIP file. If you have problems with filename encodings, please take a look at Client configuration .	Yes.
 Start workflow	<p>Start a workflow on the selected node in browser panel.</p> <div>  You can start a workflow on any node type (document, folder or mail). Optionally some workflows require you to fill a form as part of starting. </div> <div>  The available workflows are set in Administration > Profiles. </div>	No.
 Refresh	Refresh the user interface.	No.
 File uploader	<p>Bulk upload feature.</p> <div>  This feature needs to have Java enabled in your browser. More information at Bulk Uploader. </div> <div>  If you're using a recent version of Goggle Chrome, keep on mind that Java applets won't work. Please take a look at Browser configuration for more info. </div>	No.
 Restore	Restores the selected node from the trash bin.	No.
 Purge	Purges the selected node from the trash bin.	No.
 Purge trash	Cleans the user trash bin.	No.
 Exit	<p>Exits the application.</p> <div>  Users should always exit the application with this menu option and not only closing the web browser. Anyway after 30 minutes the web session of the user in </div>	No.

the server will be automatically terminated.





















Example of allowed queries on find folder, document or mail.

- Add an asterisk "*" as a placeholder for any unknown or wildcard terms, for example, "***2015**", "**2015***", "**20*15**" or "***2015***".
- Add an asterisk "?" as a placeholder for any unknown or wildcard character, for example the query "**?015**" will return occurrences where first character doesn't matter.
- The query "**2015**" is a synonym of "**2015***". On this search view application always add "*" for you at the end.

Edit

Feature	Description	Massive
Lock	Locks the selected document in a browser panel.	Yes.
Unlock	Unlocks the selected document in a browser panel. Only the user who locked the document or an user with administration credentials (force unlock) can unlock the node.	Yes.
Edit	Edits a document (check-out).	Yes.
Update	Uploads a new document version (check-in). When uploading a new version of a document, you can notify users about the document changes. The application sends the notification by mail.	No.
Cancel edit	Cancels a document edition. Only the user who edited the document or user with administration credentials (force cancel edit) can cancel edition of the node.	Yes.
Delete	Deletes selected node. When a node has been deleted, you can restore it from the trash bin. Each user	Yes.

	 <i>has its own trash.</i>	
 Copy	<i>Copies the selected node.</i>	Yes.
 Move	<i>Moves the selected node.</i>	Yes.
 Rename	<i>Renames the selected node.</i>  <i>You can cancel renaming using the ESC key.</i>	No.
 Add note	<i>Adds a note to the selected node.</i>	Yes.
 Add category	<i>Adds a category to the selected node.</i>  <i>Categories are a virtual taxonomy space that helps on the navigation of taxonomy nodes across other concepts</i>	Yes.
 Add keyword	<i>Adds a keyword to the selected node.</i>  <i>When thesaurus is enabled you can choose terms from there (SKOS).</i>	Yes.
 Merge PDF	<i>Merges two or more PDF files. The result of the process is a new PDF document.</i>  <i>Musts chooses two or more PDF files to be merged.</i> <i>You can change the process order of the PDF documents.</i> <i>You can set the name of the new PDF document result of merging the selected PDF files.</i>	No.
 Add metadata group	<i>Adds a metadata group to the selected node.</i>	Yes.

 Update metadata group	Updates a metadata group fields values of the selected node.	Yes.
 Remove metadata group	Deletes the metadata group of the selected node.	Yes.
 Add subscription	Subscribes to the selected node. <div> You can subscribe on any node type (document, folder or mail). In case you are subscribed to node of type folder, you'll be notified by mail with any change in that node and children hierarchy.</div>	No.
 Remove subscription	Un subscribe to the selected node.	No.



The document edition is based on the "**check-out / check-in**" model.


"**Check-out**" and "**check-in**" are common terms in many document management systems, including OpenKM. Their purpose is to prevent conflicts in an environment where multiple people might want to edit the same piece of content - in OpenKM a document - at the same time.














The term "**check-out**" describes the process of getting a version of a document. By checking out a document, a user can prevent others from editing that content.














The term "**check-in**" describes the process of updating a document to replace the previous version. By checking in the item, the user can allow others to edit the document, without needing to worry about overriding changes that others have made.


When a document is checked out to a user, that user can work on that document, and other users cannot. Other users must wait for the user to check in the document before they can edit. While the document is checked out, other users usually can see only the last version that was checked in; they can't see the changes that the current user has made on the document while it's been checked out.

Tools









Feature	Description	Massive
 Languages	Changes the user interface language.	No.

 Skin	Changes the user interface skin.	No.
 Debug console	Shows the frontend console.	No.
 Show administration	Shows the administration tab on a separate browser tab. <div> This option is only available for Administrators users.</div>	No.
 User configuration	Shows the user configuration. You can change your password and mail address.	No.
 OMR	Applies Object Mark Recognition on the selected node. <div> The data captured is stored in a metadata group.</div> <div> This option is only available if you have enabled some OMR template. For more information about it take a look at Administration > OMR templates.</div>	No.
 Convert	Transforms the selected document to other format. <div> Available options:<ul style="list-style-type: none">• When Save checkbox is checked a new document is created in the repository.• When Download checkbox is checked the converted document is automatically download.</div>	Yes.
 Export to Dropbox	Exports the selected node to Dropbox. <div> This option is only available if you have enabled the Dropbox feature.</div>	Yes.
 Import	Import data from Dropbox to selected folder at Navigator. <div></div>	No.

from Dropbox	 This option is only available if you have enabled the Dropbox feature.	
 Create new document	<p>Creates a new document and starts editing with Goggle Drive integration view.</p> <div>  Goggle Drive integration allows collaborative edition with several users on edition at same time. </div> <div>  This option is only available if you have enabled the Goggle Drive feature. </div>	No.
 Edit	<p>Edits the selected document with Goggle Drive integration view.</p> <div>  Goggle Drive integration allows collaborative edition with several users on edition at same time. </div> <div>  This option is only available if you have enabled the Goggle Drive feature. </div>	No.
 HTML Editor	<p>You can enable or disable integrated HTML editor.</p> <div>  This option is only available if you have enabled the HTML Editor 3 feature. </div>	No.
 Manage contacts	<p>You can add or edit existing contacts.</p> <div>  This option is only available if you have enabled the Goggle contact feature. </div>	
 Zoho edit	<p>You can editor document with integrated Zoho Editor.</p> <div>  The allowed documents are: <ul style="list-style-type: none"> • Ms Office (doc, xls) • OpenOffice and LibreOffice (odt, ods) • Text files. • CSV files. • HTML files. </div>	No.

	<ul style="list-style-type: none"> • RTF files. 	
	 This option is only available if you have enabled the Zoho feature.	







Bookmark

Feature	Description	Massive
 Home	Go to home node.	No.
 Set default home	Change default home for selected node. <div>  When user log in, the application opens in desktop view the home node of the user. By default home node is <code>"/okm:root"</code>. </div> <div>  You can select nodes in navigation and browser panels from desktop. More information at Working in desktop. </div>	No.
 Add Bookmark	Bookmark the selected node. <div>  You can select nodes in navigation and browser panels from desktop. More information at Working in desktop. </div>	No.
 Edit Bookmark	Show bookmark manager	No.
<div>  From bookmark view you can see the list of nodes added in bookmark by user. When you click on a bookmark node the application opens the node in desktop view. </div>		

Templates

Template menu dynamically build a menu hierarchy based on `"/okm:templates"` nodes. When you click on any menu option at templates, automatically starts the process of creating a new node (usually a new document based on template document).

Help

Feature	Description
 Documentation	<i>Opens a new browser tab with official openkm documentation URL.</i>
 Support request	<i>Opens a new browser tab with official community support URL.</i>
 Public forum	<i>Opens a new browser tab with official public forum URL.</i>
 Version notes	<i>Opens a new browser tab with application changelog URL</i>
 OpenKM	<i>Opens a new browser tab with official OpenKM website URL.</i>
 About OpenKM	<i>Shows a pop-up with information of application.</i>

Working in toolbar









The toolbar section has the most common features like download documents, create new folders, edit or upload files among others. The toolbar is made up of buttons with the main functions that the user will most frequently use. As with the Menu bar, occasionally the buttons will appear greyed-out, meaning that they are not available.



























The features described below are the default ones. Your toolbar can show others not described here if you have OpenKM extensions enabled.










Massive column in the tables, indicates you can check several nodes at the same time in a browser panel to apply the same action.

Feature	Description	Massive
 Find folder	<p>Find folder in the repository.</p> <div>  The search can be executed only on the context of the selected stack. </div>	No.
 Find document	<p>Finds a document in the repository.</p> <div>  The search is executed only on the context of the selected stack. </div>	No.
 Find similar document	<p>Finds similar document in the repository. The feature is based on search engine heuristic that based on contents of the selected document return similar documents.</p> <div>  The search can be executed only on the context of the selected stack. This feature is only enabled when a document is selected in a Browser view. </div>	No.
 Create folder	Creates a new folder into the selected node of the navigator panel.	No.
 Download	Downloads the selected document in a browser panel.	No.

document		
 Download document as PDF	<p>Downloads the selected document in a browser panel in PDF format.</p> <div>  <p>The feature is enabled when it is possible to convert the document to PDF format.</p> </div>	No.
 Lock	Locks the selected document in a browser panel.	Yes.
 Unlock	Unlocks the selected document in a browser panel.	Yes.
 Add document	<p>Uploads a new document into the selected node of the navigator panel.</p> <div>  <p>When the option "Import documents from ZIP" is checked you import the contents of a ZIP file respecting the hierarchy of the folder tree. If you have problems with filename encodings, please take a look at Client configuration.</p> <p>When the option "Notify to users" is checked you can send a mail notification to other application users.</p> <p>Optionally, based on Administration > Profiles configuration or Administration > Automation a wizard is shown. Wizard can require to you for setting metadata groups, categories or keywords.</p> </div>	No.
 Print	Prints the selected document in a browser panel.	No.
 Edit	Edits a document (check-out).	Yes.
 Update	Uploads a new document version (check-in).	No.
 Delete	Deletes selected node.	Yes.
 Cancel edit	Cancels document edition.	Yes.
 Add	Adds metadata group to the selected node.	Yes

metadata group		
 Delete metadata group	<i>Deletes the metadata group of the selected node.</i>	Yes.
 Add subscription	<i>Subscribes to the selected node.</i> <div>  <p>You can subscribe to any node type (document, folder or mail). In case you are subscribed to a particular node of type folder, you'll be notified by mail when any change in that node and children hierarchy occurs.</p> </div>	No.
 Remove subscription	<i>Unsubscribes to the selected node.</i>	No.
 Start workflow	<i>Starts a workflow on the selected node in a browser panel.</i> <div>  <p>You can start a workflow on any node type (document, folder or mail).</p> </div>	No.
 Refresh	<i>Refreshes user interface.</i>	No.
 Home	<i>Goes to home node.</i>	No.
 Scan	<i>Scanning.</i>	No.
 Bulk upload	<i>Bulk upload feature.</i> <div>  <p>This feature needs to have Java enabled in your browser. More information at Bulk Uploader.</p> </div>	No.
 By default	<i>Restores default panels distribution.</i>	No.

 Close tree	<i>Closes navigator panel.</i>	<i>No.</i>
 Close properties	<i>Closes properties panel.</i>	<i>No.</i>
 Expand tree	<i>Maximizes navigator panel.</i>	<i>No.</i>
 Expand explorer	<i>Maximizes browser panel.</i>	<i>No.</i>
 Expand properties	<i>Maximizes properties panel.</i>	<i>No.</i>
 Macro	<i>Executes a macro.</i> <div data-bbox="357 1151 1281 1247">  <i>This option is only available if you have enabled the Macros feature.</i> </div>	<i>No.</i>

Working in desktop

Almost all user's activity is done there. Users can work directly on folders, documents or mails.

The desktop is distributed in tree sections:

- **Navigation** panel at left hand side.
- **Browser** panel at top center.
- **Properties** panel at bottom center.

Navigation

The navigation at left hand side is an stack panel. Each stack contains a tree (folder hierarchy) structure from root that enables easily access documents and mails.

The navigation panel is divided on several stacks:

- Taxonomy.
- Categories.
- Metadata.
- Thesaurus.
- Templates.
- My documents.
- E-mail.
- Trash.

Each stack contains a tree (folder hierarchy) to help on organizing the nodes into the repository and help the user to locate fast and efficiently the data.

The **taxonomy** stack is the space for collaborative work, that means is the space where several users modify and share the company data.

The **categories** stack is a virtual taxonomy space that helps on navigation of taxonomy nodes across other concepts.



Categories is a hierarchical tree structure of category and subcategory folders. The idea is these folders and sub folders represents conceptual terms in a tree structure.

The **metadata** stack is a virtual taxonomy space that helps on the navigation of taxonomy nodes across the metadata field values.



Metadata view is a hierarchical tree structure of metadata groups. Each possible metadata value is represented

as a tree node structure.

Only fields of type Select, SuggestBox and Checkbox are displayed (free text fields like Inputs or Text area are not shown).

The **thesaurus** stack is a virtual taxonomy space that helps on the navigation of taxonomy nodes across SKOS concepts.



A **thesaurus** is a book that lists **words grouped together according to a similarity of meaning** (containing synonyms and sometimes antonyms), in contrast to a dictionary, which contains definitions and pronunciations.

SKOS concepts are linked to many, but are represented as a tree structure.

The **templates** stack is the space to set templates for all the company. That means is the space where company templates are set to be used by several users for creating new company documents.

The **My documents** stack is a private space for each user, that means that only a user is able to get access in this section and is the default where personal data is stored.

The **E-mail** stack is a private space for each user, that means only a user is able to access their data in this section and is the default place where are stored the mails of the user.

The **Trash** stack is a private space for each user, that means that only a user is able to get access in this section. When a user deletes a node it goes automatically to its own trash. When a user empty its own trash the document is definitively removed from the application.



Administrators have no restriction for accessing all repository areas, including private user sections like "My Document", "E-mail" or "Trash".

The selected node from navigator panel determines the content shown at browser panel. All the nodes shown at the browser panel have the same parent (selected node on tree).



In case of virtual stacks (categories, metadata and thesaurus) the nodes shown at browser panel do not have the same parent (selected node on tree), really they are shown there because they have a relation (category, metadata or thesaurus value) with the selected node on tree.

When you do a **right click** with the mouse button on a node at the tree is shown a **contextual menu**. The available menu options in contextual menu depending on selected stack.



A complete description of all menu options are available at: [Working in menu bar](#).

Available menu options at **Taxonomy** stack are:

- Create folder.
- Delete.

- *Rename.*
- *Move.*
- *Copy.*
- *Add document.*
- *Add bookmark.*
- *Set default home.*
- *Export to ZIP.*

Available menu options at **Categories** stack are:

- *Create folder.*
- *Delete.*
- *Move.*

Available menu options at **Templates** stack are:

- *Create folder.*
- *Delete.*
- *Rename.*
- *Move.*
- *Copy.*
- *Export to ZIP.*

Available menu options at **My document** stack are:

- *Create folder.*
- *Delete.*
- *Rename.*
- *Move.*
- *Copy.*
- *Export to ZIP.*

Available menu options at **E-mail** stack are:

- *Create folder.*
- *Delete.*
- *Rename.*
- *Move.*
- *Copy*

Available menu options at **Trash** stack are:





- Restore.
- Purge.
- Purge trash.



Metadata and Thesaurus stack view has no menu options.



Those folders to which the user is subscribed will be displayed in a different color to differentiate them:

-  You are subscribed to a folder **without subfolders** into.
-  You are subscribed to a folder **with subfolders** into.
-  You are subscribed to a **read only** folder **without subfolders** into.
-  You are subscribed to a **read only** folder **with subfolders** into.

Browser


The browser panel at the top center shows a table of nodes (documents, folders and mails) that are in the selected node of the navigation panel.

At the top of the browser panel there's a **path** that indicates the actual location in the navigation panel (for example `"/okm:root/"`). Below, there is a list of the nodes related with the selected node in the navigator panel.



Usually are listed the nodes that have the selected node as parent, except on Categories, Metadata and Thesaurus view where the nodes listed are related with the selected node and have not the same parent.

By default the browser table shows the columns:

- Status (icons display node status, for example if the node is locked is shown the **Lock**  icon).
- Checkbox (to apply actions on several nodes at same time).
- Icon (indicate node type).
- Name (name of the node).
- Title.
- Language (only for documents).
- Size.
- Update date.
- Author.
- Version (only for documents).



Also in browser table can be shown extra columns based on metadata field values. For more information about how enable it take a look at: [Administration](#) > [Profiles](#).

When you do a **right click** on the mouse button at some node it is shown a **contextual menu**. You will have distinct menu options depending on the selected node type and the visible stack panel combination.



A complete description of all menu options is available at: [Working in menu bar](#).

Available menu options when **Taxonomy, My documents** stack is visible are:

- Download.
- Edit.
- Update.
- Cancel edit.
- Lock.
- Unlock.
- Delete.
- Rename.
- Move.
- Copy.
- Convert.
- Add note.
- Add category.
- Add keyword.
- Add metadata group.
- Update metadata group.
- Merge PDF.
- Add bookmark.
- Set default home.
- Export to ZIP.
- OMR data capture.

Available menu options when **Categories, Metadata and Thesaurus** stack is visible are:

- Download.
- Edit.
- Update.
- Cancel edit.
- Lock.
- Unlock.

- *Delete.*
- *Rename.*
- *Add note.*
- *Add category.*
- *Add keyword.*
- *Add metadata group.*
- *Update metadata group.*
- *Go to (document, folder, mail). Open node location in taxonomy view.*

Available menu options when **Templates** stack is visible are:

- *Download.*
- *Edit.*
- *Update.*
- *Cancel edit.*
- *Lock.*
- *Unlock.*
- *Delete.*
- *Rename.*
- *Move.*
- *Copy.*
- *Convert.*
- *Add note.*
- *Add category.*
- *Add keyword.*
- *Add metadata group.*
- *Update metadata group.*
- *Merge PDF.*
- *Export to ZIP.*
- *Create from template.*
- *Add bookmark.*



Create from template helps you to create new documents based on existing templates. More information at: [Creating templates](#) .

Available menu options when **E-mail** stack is visible are:

- *Delete.*

- *Rename.*
- *Move.*
- *Copy.*
- *Convert.*
- *Add note.*
- *Add category.*
- *Add keyword.*
- *Add metadata group.*
- *Update metadata group.*
- *Merge PDF.*
- *OMR data capture.*

Available menu options when **Trash** stack is visible are:


- *Restore.*
- *Purge.*

The **checkbox** helps you to **select several nodes** at same time and **apply an action for all**.



You can do actions on several nodes at the same time (you have a checkbox in each row for selecting the node). The massive actions allowed are:

- *Delete.*
- *Move.*
- *Copy.*
- *Add notes.*
- *Add categories.*
- *Add keywords.*
- *Add metadata groups.*
- *Change metadata group field values.*






On **making a subscription** to node an  icon will appear at the side of node indicating that **you are subscribed**.



When there is no checkbox selected the action will be applied to the selected node (blue row mark the selected node in browser view).

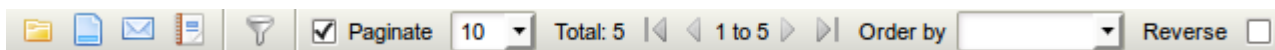
The  Massive icon displays a contextual menu for doing quick actions:








Feature	Description


 Select all	Selects all the nodes.
 Select all folders	Selects all the nodes of type folder.
 Select all document	Selects all the nodes of type document.
 Select all mails	Selects all the nodes of type mail.
 Remove selection	Unselect all the nodes.

Pagination

When **pagination feature** is enabled is shown and extra toolbar:



Feature	Description
 Show / hide folders	Shows / hides nodes of type folder in the browser panel.
 Show / hide documents	Shows / hides nodes of type document in the browser panel.
 Show / hide mails	Shows / hides nodes of type mail in the browser panel.
 Filter	Filters nodes.
Paginate checkbox	Enables or disables the pagination feature.
Limit select	Sets the limit of nodes to be shown at the same time.
 Go to the beginning	Goes to the beginning of the list.
 Go to previous	Goes to previous list.
 Go to next	Goes to next list.

 Go to the end	<i>Goes to the end of the list.</i>
Order select	<i>Sets the column value for that ordering.</i>
Reverse checkbox	<i>Reverses the ordering.</i>

Mouse features

- **Double-click** on a node of type folder to **open the folder**.
- **Double-click** on a node of type document to **download the document**.
- **Right click** on a node will show a **contextual menu**.
- **Drag & drop**. You can drag & drop nodes keeping the left mouse button held down and moving to destination node in the Navigator panel.

Properties

The properties panel at bottom center, show a collection of tabs that contains the grouped properties or features like preview associated with the selected node. The selected node can be a navigator node or a browser node.







Basically the selected node at the navigator panel determines the content shown at the browser panel and depending if the last selected node by the user is at navigator or browser panel the properties panel will shown the properties of the active node at navigator or browser.







The features described below are the default features. In your properties tab can be shown others not described here if you have OpenKM extensions enabled.

Tabs on nodes of type document

Tab	Description
Properties	<p>Displays properties of the selected document.</p> <p>From this view you can:</p> <ul style="list-style-type: none"> • Set the keywords of the document. • Change the title of the document. • Set relations between the document and categories. <div>  From this tab is shown a list with the users subscribed to the node. </div>

	<p>For a quick node sharing, there are two special icons named URL and WebDAV when clicking on them, it will copy in your clipboard a browser or WebDAV URL.</p>
Notes	<p>Displays a list of notes related with the document. From this view you can add, edit or remove your own notes.</p> <p> Administrators can manage notes of other users.</p>
History	<p>Displays a list with the versions of the document.</p> <p>From this view you can:</p> <ul style="list-style-type: none"> • Restore to a previous version of the document. • Preview any document version on the right panel. • Compare two versions of the document on the preview panel. • Compact the history. <p> The action of compacting the history will remove all the document version except the active. This action freezes repository size but must be take in consideration that this action can not be reverted.</p>
Security	<p>Displays the security grants of the document. From this view you can change the security grants.</p> <p> • Click on Update button to change the security grants.</p> <p>When "Recursive permission changes" checkbox is checked the changes will be applied recursively.</p> <p>You should also be interested on reading:</p> <ul style="list-style-type: none"> • Security configuration. • Security configuration parameters. • "Advanced filter" option on Profile General tab.
Preview	<p>Displays a pre visualization of the document.</p>
Activity	<p>This view performs a complete audit trail of the document. From this view you can filter by specific actions.</p>




log	<p>When the checkbox named View is not checked, the most common actions like GET_CHILDREN_DOCUMENTS are automatically filtered.</p> <div data-bbox="336 347 1404 488">  <p>The reason why most common actions are filtered when checkbox name View is not checked is that these events usually create a large list of actions and often are not relevant from the audit trail point of view.</p> </div> <div data-bbox="336 510 1404 600">  <p>This option is only available if you have enabled the Activity log feature.</p> </div>
Forum	<p>Displays a forum associated with the document.</p> <p>From this view you can:</p> <ul style="list-style-type: none"> • Add a new topic. • Post a reply of the topic. • Quote a reply of the topic. • Edit a reply of the topic. • Delete a topic. <div data-bbox="336 1061 1404 1187">  <p>To delete or edit a post of the topic you must be the owner or have administrator grants.</p> <p>This option is only available if you have enabled the Forum feature.</p> </div>
Wiki	<p>Display a wiki associated with the document.</p> <p>From this view you can:</p> <ul style="list-style-type: none"> • Create a wiki. • Update a wiki. • Show the history of wiki versions. • Restore a version of the wiki. <div data-bbox="336 1621 1404 1711">  <p>This option is only available if you have enabled the Wiki feature.</p> </div>
Workflow	<p>Display the workflows associated with the document.</p> <p>From this view you can:</p> <ul style="list-style-type: none"> • Show the detail of the workflow variables.





- **Show a graph** where it is marked the actual node.



This option is only available if you have enabled the **Workflow** feature.

Tabs on nodes of type folder




Tab	Description
Properties	<p>Displays properties of the selected folder.</p> <p>From this view you can:</p> <ul style="list-style-type: none"> • Set the keywords of the folder. • Set relations between the folder and categories. <div>  <p>This tab shows a list with the users subscribed to the node.</p> <p>For a quick node sharing, there are two special icons named URL and WebDAV when clicking on them, it will copy in your clipboard a browser or WebDAV URL.</p> </div>
Notes	<p>Displays a list of notes related with the folder. From this view you can add, edit or remove your own notes.</p> <div>  <p>Administrators can manage other user's notes.</p> </div>
Security	<p>Displays the security grants of the folder. From this view you can change the security grants.</p> <div>  <ul style="list-style-type: none"> • Click on Update button to change the security grants. <p>When the "Recursive permission changes" checkbox is checked the changes will be applied recursively.</p> <p>You could be interested on reading:</p> <ul style="list-style-type: none"> • Security configuration. • Security configuration parameters. • "Advanced filter" option on Profile General tab. </div>
Activity log	<p>This view performs a complete audit trail of the folder. From this view you can filter by specific actions.</p> <p>When the checkbox named View is not checked, the most common actions like</p>






	<p>GET_CHILDREN_DOCUMENTS are automatically filtered.</p> <div>  <p>The reason why most common actions are filtered when checkbox name View is not checked is that these events usually create a large list of actions and often are not much relevant from the audit trail point of view.</p> </div> <div>  <p>This option is only available if you have enabled the Activity log feature.</p> </div>
Forum	<p>Display a forum associated with the folder.</p> <p>From this view you can:</p> <ul style="list-style-type: none"> • Add new topic. • Post a reply to the topic. • Quote a reply tof the topic. • Edit a reply of the topic. • Delete a topic. <div>  <p>To delete or edit a post of the topic you must be the owner or have administrator grants.</p> <p>This option is only available if you have enabled the Forum feature.</p> </div>
Wiki	<p>Displays a wiki associated with the folder.</p> <p>From this view you can:</p> <ul style="list-style-type: none"> • Create a wiki. • Update a wiki. • Show the history of wiki versions. • Restore a version of the wiki. <div>  <p>This option is only available if you have enabled the Wiki feature.</p> </div>
Workflow	<p>Display the workflows associated with the folder.</p> <p>From this view you can:</p> <ul style="list-style-type: none"> • Show the detail of the workflow variables. • Show a graph where is marked the actual node.



This option is only available if you have enabled the **Workflow** feature.

Tabs on nodes of type mail

Tab	Description
Properties	<p>Display properties of the selected mail.</p> <p>From this view you can:</p> <ul style="list-style-type: none"> • Set the keywords of the mail. • Set relations between the mail and categories. <div>  <p>This tab shows a list with the users subscribed to the node.</p> <p>For quick node sharing, there are two special icons named URL and WebDAV when clicking on them, it will copy in your clipboard a browser or WebDAV URL.</p> </div>
Notes	<p>Displays a list of notes related with the mail. From this view you can add, edit or remove your own notes.</p> <div>  <p>Administrators can manage notes of other users.</p> </div>
Security	<p>Displays the security grants of the mail. From this view you can change the security grants.</p> <div>  <ul style="list-style-type: none"> • Click on Update button to change the security grants. <p>When "Recursive permission changes" checkbox is checked the changes will be applied recursively.</p> <p>You might be interested on reading:</p> <ul style="list-style-type: none"> • Security configuration. • Security configuration parameters. • "Advanced filter" option on Profile General tab. </div>
Preview	<p>Displays a pre visualization of the mail.</p> <div> <p>From this view you can also download the mail attachments or copy them to a destination</p> </div>

	 <i>folder.</i>
Activity log	<p>This view performs a complete audit trail of the mail. From this view you can filter by specific actions. When the checkbox named View is not checked, the most common actions like GET_CHILDREN_DOCUMENTS are automatically filtered.</p> <div>  The reason why most common actions are filtered when checkbox name View is not checked is that these events usually create a large list of actions and often they are not relevant from the audit trail point of view. </div> <div>  This option is only available if you have enabled the Activity log feature. </div>
Forum	<p>Displays a forum associated with the mail.</p> <p>From this view you can:</p> <ul style="list-style-type: none"> • Add a new topic. • Post a reply to the topic. • Quote a reply of the topic. • Edit a reply of the topic. • Delete a topic. <div>  To delete or edit a post of the topic you must be the owner or have administrator grants. This option is only available if you have enabled the Forum feature. </div>
Wiki	<p>Displays a wiki associated with the mail.</p> <p>From this view you can:</p> <ul style="list-style-type: none"> • Create a wiki. • Update a wiki. • Shows the history of wiki versions. • Restore a version of the wiki. <div>  This option is only available if you have enabled the Wiki feature. </div>

Workflow

Display the workflows associated with the mail.

From this view you can:

- Show the **detail** of the workflow variables.
- **Show a graph** where is marked the actual node.



This option is only available if you have enabled the **Workflow** feature.

Shortcuts

Shortcuts can be used from desktop view.

Available shortcuts:

Shortcut	Description
F2	Rename a node.
CTRL+SHIFT+C	Copy a node.
CTRL+SHIFT+X	Cut a node.
CTRL+SHIFT+V	Paste a node.
CTRL+D	Downloading.
CTRL+G	Show UI testing popup.
CTRL+Z	Show UI log.
CTRL+N	Create folder.
F5	Refresh desktop view.

<i>INSERT</i>	<i>Add new document.</i>
<i>ALT+B</i>	<i>Bookmark a node.</i>
<i>F1</i>	<i>Show help.</i>
<i>CTRL+ALT+Q</i>	<i>Exit application.</i>
<i>ALT+DELETE</i>	<i>Delete a node</i>

Working in search

The search tab enables the query the repository. The search view is distributed in four sections:

- Searches panel.
- Search panel.
- Control panel.
- Results.

Searches panel

The Search panel at left hand side is an stack panel.







Once a search has been done, it can be saved for future sessions by entering a name that identifies it and clicking on the "Save search" button.

This panel have two stacks:

- Stored searches.
- User news.

Stored searches shows a list of user's searches that has been saved to be use later. With a double-click on stored search, you can execute the search again. With right click on a stored search will be shown a contextual menu with this options:

Feature	Description
 Run	Run stored search.
 Delete	Delete an stored search.
 Share query	Share an stored search.
 This option is only available if you have enabled the Messaging feature.	



The stored searches and users news are only visible for the user who created them.

User news shows a list of user searches that are used as push service (usually executed every 30 minutes). These searches are periodically executed and the results are shown at **Dashboards > News** panel.



User news helps users to be periodically updated about repository news based on his own queries.

Will be shown only new results that have appeared since the last time the query was executed.

Search panel

This panel at the top center gives access to different searches to query the content of the application:

- **Default search** is a full-text search. Default search is also available from the top right corner on the screen.
- **Advanced search** enables you to search on keywords, metadata and additional criteria such as location in the application.

You can switch between default or advanced search checking or un checking the "**View advanced mode**" checkbox.



Default search engine configuration uses **stemming** and **stopwords**.

- **Stemming** is the term used in linguistic morphology and information retrieval to describe the process for reducing inflected (or sometimes derived) words to their word stem, base or root form generally a written word form. The stem needs not to be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. For instance, if you type "fishing" in a full-text search field, search results will return documents with "fishing", "fish", "fisher". In that case, the stem is "fish".
- **Stopwords** usually refer to the most common words in a language, there is no single universal list of stop words used by all processing of natural language tools, and indeed not all tools even use such a list. The stopword are removed from search. For example "the", "is", "at", "which" and "on" are stopwords.

Operators and wildcards

By default internally the search engine use the "**AND**" operator.

You can use wildcards:

- Add an asterisk "*" as a placeholder for any unknown or wildcard terms, for example, "***2015**", "**2015***", "**20*15**" or "***2015***".
- Add an asterisk "?" as a placeholder for any unknown or wildcard character, for example the query "**?015**" will return occurrences where first character doesn't matter.
- The query "**2015**" is a synonym of "**2015***". On this search view application always add "*" for you at the end.

Default search

Default search is a full-text search.



Default search shows results nodes only in `"/okm:root"` context, results from other context are automatically filtered.



The search takes in consideration:

- Content (document content).
- Title.
- Keywords.
- Name.
- Subject (mail subject).
- Body (mail content)
- Notes.
- UUID (node UUID).



Always at the top right corner you have quick access with Quick search box feature.

Advanced search



Advanced search enables you to search on keywords, metadata and additional criteria such as location in the application.

The advanced search organizes the query fields on tree tabs:


- Basic (most common fields).
- Advanced.
- Metadada.





The available criteria from basic view :

Field	Description	Wildcard allowed
Context	<p>Sets the context where is executed the query.</p> <p>Available context:</p> <ul style="list-style-type: none"> • Taxonomy. • Templates. • My documents. • E-mail. • Trash. 	No.


Content	<p>The content of the node.</p> <p><i>Document content or mail body.</i></p>	Yes.
Name	The name of the node.	Yes.
Title	The title of the node.	Yes.
Keyword	The associated keywords of the node.	Yes.
Language	<p>The language of the document.</p> <p> Only the nodes of typed document has language value.</p>	Yes.
User	The creator.	No.
Date range	<p>The range of dates where the node has been modified (or created).</p> <p> Each selected date is set to 00 hours, 00 minutes, 00 seconds and 00 milliseconds. That means if you want to do a query of a node modified the day 2015-08-15 must set a range from "2015-08-15" to "2015-08-16" (internally you're querying from "2015-08-15 00:00:00" to "2015-08-16 00:00:00").</p> <p>It is not mandatory to set both dates for range ("from" or the "to"), you can set only one date.</p>	No.

The available criteria from advanced view :

Field	Description	Wildcard allowed
Folder	<p>Restrict results to some folder.</p> <p> When Recursive checkbox is checked also considers the subfolders.</p>	No.

Category	Restricts results to some category. The nodes must be associated to a specific category.	No.
Type	Restrict the types of node: <ul style="list-style-type: none"> • Document • Folder • Mail <div>  When Mail checkbox is checked are shown extra fields (from, to , subject) </div>	No.
MIME type	Set the document MIME type.	No.
Notes	The content of the notes.	Yes.
From	The from mail field. <div>  This field is only available if you have checked the Mail checkbox. </div>	Yes.
To	The to mail field. <div>  This field is only available if you have checked the Mail checkbox. </div>	Yes.
Subject	The subject mail field. <div>  This field is only available if you have checked the Mail checkbox. </div>	Yes.

From metadata view you can filter by any metadata field from existing metadata groups:

- Click on  **Add metadata group** button.
- **Choose one metadata group** from the list.
- **Choose a property** from the list.






- Click  **Add** button.







On almost input and text area fields you can use wildcards.

Control panel

Control panel at the right top corner has several features:

Featureright	Description
Switch to Advanced mode checkbox	When checked is shown the advanced mode search view.
Results compact view checkbox	When checked the results are shown in a compact table with minimal details.
View metadata groups checkbox	When checked are shown metadata groups associated to each node in results view. <div> This option is only available when "Results compact view" is not checked.</div>
Save as user news checkbox	When checked if the user saves a search, will be set to user news search.
 Save search button	Once a search has been done, it can be saved for future sessions by entering a name that identifies it and clicking on the "Save search" button. <div> Depending if "Save as user news" checkbox is checked or not will be saved as "stored search" or "user news".</div>
Page results lists	Set the limit of nodes to be shown in the results list.
 Clean button	Resets the search.
 Search	Executes the search.

Results	<i>Indicates the number of results (total).</i>
 Go to previous	<i>Goes to previous list.</i>
 Go to next	<i>Goes to next list.</i>
 Export to CSV	<i>Downloads a CSV file with results.</i>
 Export to ZIP	<i>Downloads a ZIP file with results.</i>

Results






The Results panel at the bottom center shows a table of nodes (documents, folders and mails) what are the result of querying the repository. Results panel are shown in:






- Compact view (a table with a single row for each node).
- Full view (a detailed description for each node).







Stars indicates the relevance of the document found. This value is established by the Search engine according to the search field and the results found. The list of documents found is ordered from most to least relevant.

When you do a right click on compact view will be shown a contextual with the options:

Feature	Description
 Download	<i>Downloads the node.</i> <div>  <i>Only shown in document or mail nodes.</i> </div>
 Go to document	<i>Goes to the document location.</i> <div>  <i>Only shown in document nodes.</i> </div>
 Find similar documents	<i>Find similar document in the repository. The feature is based on search engine heuristic that based on contents of the selected document return similar documents.</i>

	 Only shown in document nodes. The search is executed only on the context of the selected stack. This feature is only enabled when is selected a document in Browser view.
 Go to folder	Goes to folder location.  Only shown in folder nodes.
 Go to e-mail	Goes to mail location.  Only shown in mail nodes.

From full view you see detailed information of each node and also are available some features:

Feature	Description
 Find similar documents	Finds similar documents in the repository. The feature is based on search engine heuristic that based on contents of the selected document return similar documents.  Only shown in document nodes. The search be executed only on the context of the selected stack. This feature is only enabled when is selected a document in Browser view.
 Download	Download the node.  Only shown in document or mail nodes.

Working in dashboard

The dashboard panel provides a set of dashboards, each user has a set of personal dashboard that displays the information relevant for him and helps to have a global view of the application activity.

The dashboard panel :

- **User view.** Displays the actions that have been carried out in the system
- **General view.** Displays a global view of application activity.
- **News.** Displays information based on the scheduled user search queries.
- **Mail.** It's a virtual navigations across all the repository of mails.
- **Workflow.** Users work in company business process tasks.
- **Keyword map.** Displays a tag cloud based on all the repository keywords.
- **Messaging.** Displays internal OpenKM messaging.
- **Forum.** Displays a forum.
- **Wiki.** Displays a wiki.
- **Meeting.** Manages meetings with OpenMeetings.



Messaging is only available if you have enabled the **Messaging** feature.

Forum is only available if you have enabled the **Forum** feature.

Wiki is only available if you have enabled the **Wiki** feature.

Meeting is only available if you have enabled the **OpenMeetings** feature.




The results listed in dashboard boxes depends on the security level access for each user.

User view

In the user's information view, users can access some interesting queries about the last user operations.

The information is organized in several boxes:


Box	Description
Locked documents	Displays a list with the actual documents locked by the user.
Edited documents	Displays a list with the actual documents checkout by user.
Last download documents	Displays a list with the last 20 documents downloaded by the user.

Subscribed documents	<i>Lists subscriptions into documents by the user.</i>
Subscribed folders	<i>Lists subscriptions into folders by the user.</i>
Last modified documents	<i>Displays a list with the last 20 documents modified by the user.</i>
Last created documents	<i>Displays a list with the last 20 documents created by the user.</i>
 <i>Each box indicates the number of new items pending to be viewed by the user.</i> <i>When a user clicks on a row, application will open up automatically the repository destination node.</i>	

General view

In the general information view, users can access some interesting queries about the last repository operations done by all users.

The information is organized in several boxes:

Box	Description
Last week top view documents	<i>Displays a list with the last 20 documents most viewed during last week.</i>
Last month top viewed documents	<i>Displays a list with the last 20 documents most viewed during last month.</i>
Last week top modified documents	<i>Displays a list with the last 20 documents most modified during last week.</i>
Last month top modified documents	<i>Displays a list with the last 20 documents most modified during last month.</i>
Last created documents	<i>Displays a list with the last 20 documents created.</i>
Last modified documents	<i>Displays a list with the last 20 documents modified.</i>
 <i>Each box indicates the number of new items pending to be viewed by the user.</i>	


When a user clicks on a row, the application will open up automatically the repository destination node.

Mail

In the mail information view, users can access some interesting queries about the last mails operations done.

The information is organized in several boxes:

Box	Description
Electronic mails	Displays a list with the last 20 mails created.
Attachments	Displays a list with the last 20 attachments created.



Each box indicates the number of new items pending to be viewed by the user.

When a user clicks on a row, the application will open up automatically the repository destination node.

News

The News view display information based on scheduled user search queries. Each query is shown into a separate box. As a push service (usually executed every 30 minutes) the results are refreshed.



More information about registering "Users news" at: [Working in search](#).

Workflow

In the Workflow panel are shown the workflows on execution. The Workflow panel is organized in two sections:

- Workflow lists.
- Workflow detail.

Workflow lists

In the Workflow lists panel are shown the workflows on execution organized in two boxes:

- Pending tasks.
- Unassigned pending task.



The workflows at "Pending task" box has yet to be assigned to the logged user and are waiting for him.

The workflows at "Unassigned pending task" box has not yet beng assigned to a single user. There is a group of users who can auto assign the workflow themself (one of them is the logged user). The workflow is waiting for this auto asignation.

Workflow detail

In the Workflow detail panel it is shown the workflow details. From this view you can see low level information data like "process instance", "process definition" or "data" and also **go to the next workflow task completing the workflow form**.

To complete a workflow task:

- **Click on workflow at Pending task box.** It will be shown the workflow form.
- **Fill in the workflow form.**
- **Click on any of the available buttons** to submit the values.

Keyword map




The Keyword map view shows a tag cloud based on all the repository keywords. The Keyword map is organized in four sections:


- **Toolbar.**
- **Browser.**
 - **Tag cloud.**
 - **Results.**
- **Keywords list.**
- **Results controller.**

Toolbar

The toolbar is on top.







Feature	Description
 Low detail	Displays result nodes in low detail.
 Medium detail	Displays result nodes in medium detail.
 Full detail	Displays result nodes in full detail.

Keywords	Shows keywords used for filtering.
Keyword selector	The suggestbox selector helps to quickly choose the keywords for filtering.
 Clean	Reset Keyword map view.


Browser


The Browser panel is at the center. The panel is divided in two sections, at the top the **Tag cloud** (visual representation of text data) and at the bottom the **Results** table within the nodes that comply with the filtering rules.


Available features on results :

-  **Go to document.**
-  **Go to folder.**
-  **Go to e-mail.**

In the "Medium" and "Full" detail view each node display its own associated keywords in two colors blue (for what are used on filtering) and grey.

Clicking on  blue keyword you will remove from filtering rules .

Clicking on  grey keyword you will add to filtering rules .

Keywords lists

The Keywords lists is at the right side. The keywords list is an stack panel organized in tree stacks:




- Related keywords.
- Top Keywords.
- All Keywords.

In the "Related keywords" stack are listed the related keywords based on the actual filtering rules. In the "Top Keywords" stack are listed the most commonly used keywords and finally in "All keywords" are list all the keywords.

Results controller

The results controller is at te bottom.

Feature	Description
Context selected	Choose the context.

	 Available context are: <ul style="list-style-type: none"> • Taxonomy. • Templates. • My documents. • E-mail. • Trash. • All repository.
Page results	Set the limit of the nodes to be shown at the same time.
Results	Displays the number of results.
 Go to previous	Go to previous list.
 Go to next	Go to next list.

Messaging

Messaging helps you proposing a nodes subscription to the users, sharing queries and sending text messages between users.

The Message panel is organized in three sections:

- Message type
- Browser
- Preview


Message type

Message type is on left side. Message type is an stack panel with the following options:

- Propositions.
- Proposed queries.
- Messages received.
- Messages sent.


Feature	Description


Propositions	<i>Displays users who have proposed subscriptions to nodes.</i>
Proposed queries	<i>Displays users who have shared their search queries.</i>
Messages received	<i>Displays users who have sent messages to the user.</i>
Messages sent	<i>Displays users who you have sent messages.</i>


Bold users indicate you have pending information to be reviewed.

Browser





Browser panel is on the top right. The panel list the messages based on "Message type" selection.


Bold message indicate you have still not previewed.

Right clicking on a message will show a contextual menu with  **Delete** action.








Preview

Preview panel is on the bottom right. In this panel the message is previewed.

Feature	Description
Subscribe	<i>Subscribes to a node.</i> <div>  Available when "Proposition" stack is visible. </div>
 Execute search	<i>Executes shared search.</i> <div>  Available when the "Proposed queries" stack is visible. </div>
Accept	<i>Accepts a shared search.</i> <div>  Available when "Proposed queries" stack is visible. </div>





Forum

From the forum view you can **manage all discussions** on nodes and also **create new forums** not linked with any node.

Feature	Description	Need administrator grants
 New forum	Adds a new forum.	Yes.
 Edit forum	Edits a forum.	Yes.
 New topic	Adds a new topic to an existing forum.	No.
 Post reply	Posts a reply in an existing topic.	No.
 Quote	Quotes a posting.	No.
 Edit post	Edits a posting.	No when you are the owner.
 Delete	Deletes a post.	No when you are the owner.

Wiki

Display the main wiki page.

Field / Property	Description
 Update	Edits wiki contents.
 History	<p>Wiki page history.</p> <div>  Available options: <ul style="list-style-type: none"> • View older page. • Restore previous version. </div>
 Unlock	Cancels wiki edition.


**Delete***Deletes wiki page.*

Working in status bar

The **Status** panel at the bottom provides general information. The status panel is organized in:

- OpenKM news alert.
- The connected user.
- User repository size.
- Chat.
- Alerts.
- HTML documents on edition.
- Message panel.

OpenKM news alert

When there are OpenKM news is shown a blink  **alert icon** in the left side.



Click on  **OpenKM news icon** to display the news popup.

The connected user

In left side is shown Username.

User repository size

In the user repository size is shown the total repository size of the user (documents and mails owned by the user).

Chat



You can  **connect** and  **disconnect** user from internal application chat.













By default Chat feature is disabled. More information about Chat feature at [Administration > Profiles](#) .

Alerts

Click on any icon for opening related dashboard view.

Alert	Description
 Locked nodes	Displays the number of locked nodes by user.
 Document edited	Displays the of edited documents by user.

 Subscription	Displays the number of nodes subscribed by user.
 User news	Displays the number of users news pending to be reviewed by user.
 Workflow tasks	Displays the number of workflows tasks pending by user.
 Workflow pooled tasks	Displays the number of workflows pooled tasks pending to be assigned.
 Proposed subscriptions	<p>Displays the number of proposed subscriptions by other users.</p> <div>  This option is only available if you have enabled the Messaging feature. </div>
 Shared queries	<p>Displays the number of shared queries pending to be approved by the user.</p> <div>  This option is only available if you have enabled the Messaging feature. </div>
 Internal messages	<p>Displays the number of internal messages sent by other users pending to be reviewed by the user.</p> <div>  This option is only available if you have enabled the Messaging feature. </div>

Message panel

In the right side are shown application text messages (for example while previewing a document are shown messages about the preview stages).

Bulk Uploader

The Bulk upload feature helps you to insert or update multiple documents at the same time without having to go to Add new document and select each document individually.



The feature is available from Menu bar (see [Working in menu bar](#)) and Toolbar (see [Working in toolbar](#)).

Uploader detects when the document to upload already exists in the repository, in that case, it will ask you if want to add this new content as a new document version.




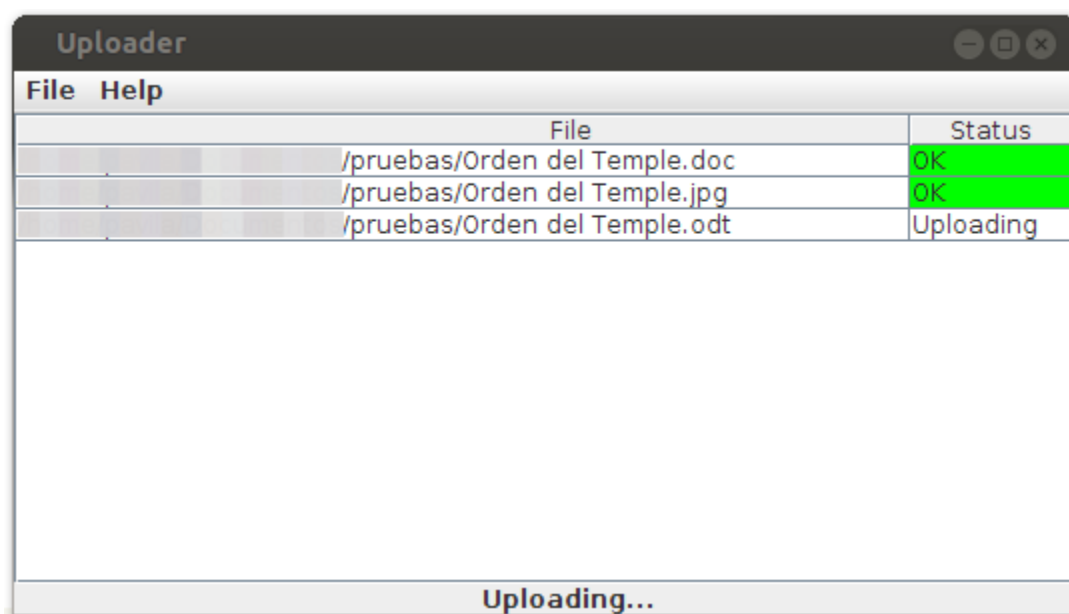
This feature needs to have Java enabled in your browser.



If you're using a recent version of Goggle Chrome, keep on mind that Java applets won't work. Please take a look at [Browser configuration](#) for more info.

Usage

- **Choose node** (folder or record) on your repository.
- Click on  **Bulk upload icon**. Will be shown the Uploader Window.



- **Drag and drop files or folders** from your desktop computer **to the Uploader** window. The files and folders will be automatically uploaded to OpenKM, to your currently selected folder path



If a problem occurs while uploading a document, it will show an **"Error"** on the status column.

HTML editor extension









The application comes with an integrated HTML editor.





This option is only available if you have enabled the **HTML Editor 3** feature.

You can edit several HTML documents at the same time and switch the editor between them for copying and paste operations.

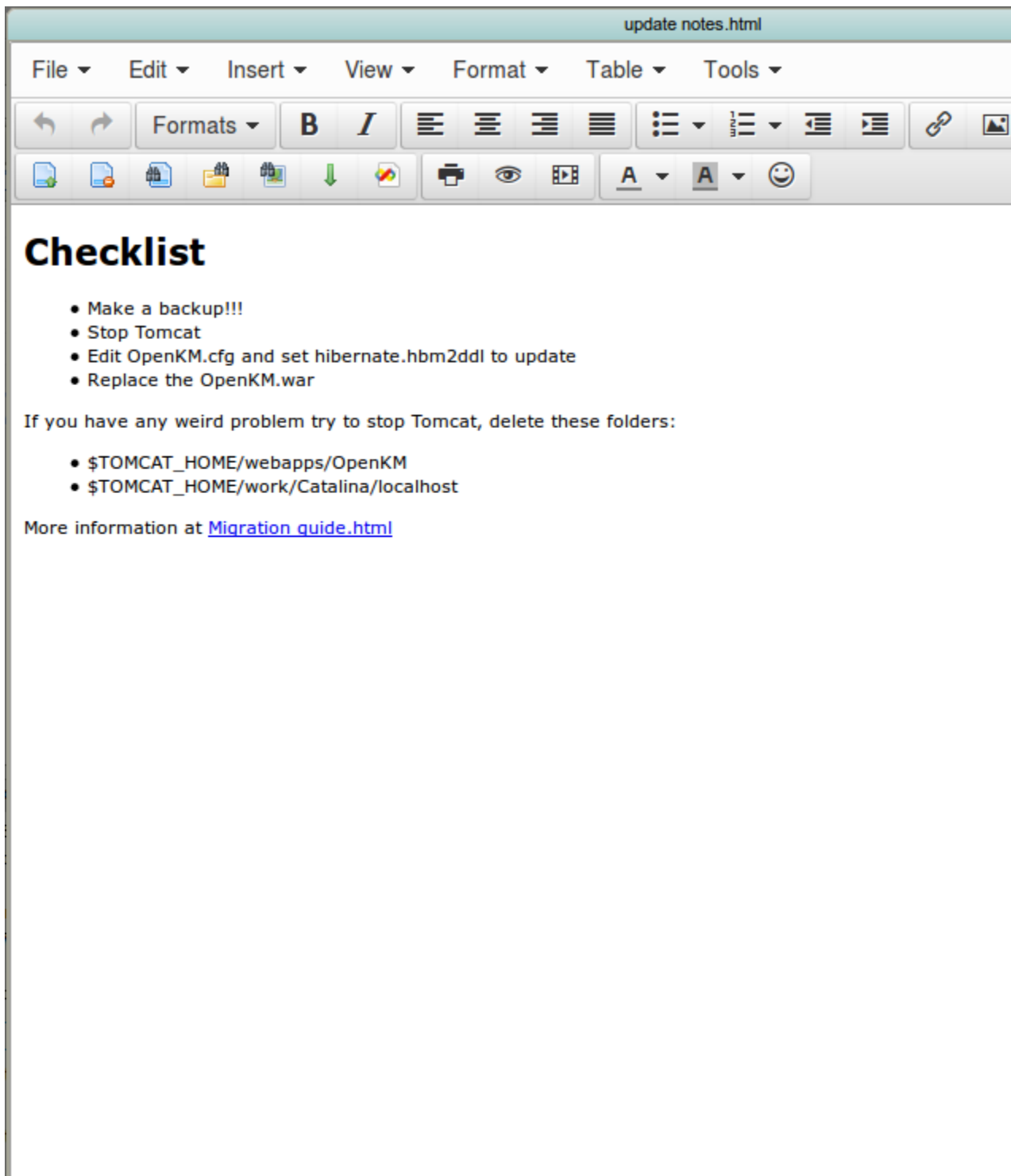
Features

Feature	Description
 Update	Upload a new document version (check-in). <div>  When uploading a new version of a document, you can notify users about document changes. The application send the notification by mail. </div>
 Cancel edit	Cancel document edition.
 Link document	Add link to a document.
 Link folder	Add link to a folder.
 Add image	Add image.
 Code	Add code section.
 You can navigate across linked documents and folders or download documents from Preview view.	

Edit HTML

- **Choose *HTML* document.**
- Click on  **Edit icon.**
- **Modify the content.**
- Click on  **Update icon.**

Editor panel



Preview panel

◀ ▶ update notes.html

Checklist

- Make a backup!!!
- Stop Tomcat
- Edit OpenKM.cfg and set hibernate.hbm2ddl to update
- Replace the OpenKM.war

If you have any weird problem try to stop Tomcat, delete these folders:

- \$TOMCAT_HOME/webapps/OpenKM
- \$TOMCAT_HOME/work/Catalina/localhost

More information at [Migration guide.html](#)


Extratab extension

The ExtraTab allows to easily add new tabs with a name and specific URL.



This option is only available if you have enabled the **Extratab** feature.

Configuration

Field / Property	Type	Description
<i>extra.tab.workspace.label</i>	String	Label of the tab. <div>Some text</div>
<i>extra.tab.workspace.url</i>	String	The URL of the tab. <div>http://www.openkm.com</div> <div> If URL does not start with http it's considered as OpenKM relative path and will be merged with your OpenKM server URL like YOUR_OPENKM_URL/url</div>



More information at [Configuration parameters](#).

Sample

The screenshot shows the website of the Universitat de les Illes Balears (UIB) in a web browser window. The browser's address bar is empty, and the page title is "Universitat de les Illes Balears". The website has a blue header with the UIB logo and navigation links: "Inici", "La UIB", "Alumnes", "Estudis", "Postgrau", "Mobilitat", "Recerca", and "Empreses". Below the header, there is a main content area with a large banner for "AJUDA'NS A MILLORAR" (Help us improve) featuring a landscape image and the hashtag #SOM UIB. To the left of the banner, there is a section titled "Ara interessa" (What interests you now) with a list of links: "Alumnes col·laboradors", "Ajuda'ns a millorar la UIB", "Beques Santander", "Seràs UIB", and "Continua amb la UIB". Below the banner, there is a section titled "Actualitat" (News) with three articles: "02/12/2013 Roda informativa: L'Obra Social 'la Caixa' i la Universitat de les Illes Balears firmen dos convenis per fomentar les activitats de recerca i la transformació del model econòmic", "29/11/2013 La UIB analitza les darreres tendències en comercialització turística", and "29/11/2013 Es presenta la 'Guia de bones pràctiques'". To the right of the news section, there is a section titled "Agenda" (Calendar) with three events: "29/11/2013 II Fòrum sobre Intermediació Turística: reptes de futur", "29/11/2013 DEA Economics Seminar Series: Non-exponential mixtures, non-monotonic financial hazard functions and t...", and "29/11/2013 COLÒMBIA: claus per desxifrar". The browser's status bar at the bottom shows "Connected as Administrator", "71.0 KB", and various system icons.

File Edit Tools Bookmarks Templates Help

Desktop

Universitat de les Illes Balears

UIBdigital | Webmail | Portal PAS-PDI | Directori | Mapa del v

Inici La UIB Alumnes Estudis Postgrau Mobilitat Recerca Empreses

Ara interessa

- > Alumnes col·laboradors
- > **Ajuda'ns a millorar la UIB**
- > Beques Santander
- > Seràs UIB
- > Continua amb la UIB

Ajuda'ns a millorar la UIB

Campanya per premiar estudiants que proposin accions i idees per millorar la UIB

Actualitat

02/12/2013 Roda informativa: L'Obra Social "la Caixa" i la Universitat de les Illes Balears firmen dos convenis per fomentar les activitats de recerca i la transformació del model econòmic

29/11/2013 La UIB analitza les darreres tendències en comercialització turística

29/11/2013 Es presenta la "Guia de bones pràctiques"

Agenda

29/11/2013 II Fòrum sobre Intermediació Turística: reptes de futur

29/11/2013 DEA Economics Seminar Series: Non-exponential mixtures, non-monotonic financial hazard functions and t...

29/11/2013 COLÒMBIA: claus per desxifrar

Connected as Administrator 71.0 KB

Forum extension

The Forum extension helps you creating discussion topics on nodes (documents, folders, mails or records). Also there is a general purpose forum available from Dashboard view.



This option is only available if you have enabled the **Forum** feature.

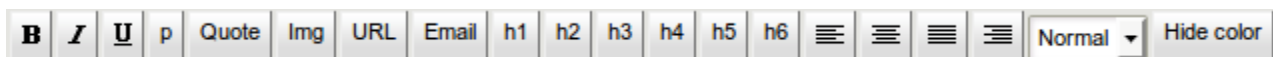


Only administrators can create, update or delete forums from Dashboard view.


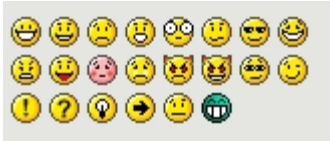





Only administrators can view the "document discussions" forum in Dashboard view. This forum contains all the node topics.

Nobody can delete "document discussions" forum in Dashboard view.

Editor features



Feature	Description
B	<i>Bold.</i>
I	<i>Italic.</i>
U	<i>Underline.</i>
p	<i>New paragraph.</i>
Quote	<i>Make a quotation.</i>
Img	<i>Add image url.</i>
Email	<i>Add email.</i>
h1 to h6	<i>Header settings.</i>
Align Left, center, justify and right.	<i>Set text justification.</i>

Text size	Set text size
Show colour	Palette of colors. 
Smilies	Smilie icons. 
	Add document link.
	Add image.
	Add folder link.
	Add mail link.
	Add record link.

Create new topic


- **Choose a node** (document, folder, mail or record).
- Click on **New topic button**.
- **Fill the form**.
- Click on **Create button**.

Create reply


- **Choose a topic on a node**.
- Click on **Post reply button**.
- **Fill the form**.

- Click on **Create button**.

Update post

- Choose a topic on a node.
- Click on  **Edit icon**.
- **Modify the form**.
- Click on **Update button**.

Delete post

- Choose a topic on a node.
- Click on  **Delete icon**.
- Click on **Accept button**.

Create forum




This option is available in Dashboard view only for administrators .

- Click on **New forum button**.
- **Fill the form**.
- Click on **Create button**.

Update forum



This option is available in Dashboard view only for administrators .

- Click on  **Edit icon**.
- **Modify the form**.
- Click on **Update button**.

Delete forum



This option is available in Dashboard view only for administrators .

- Click on  **Delete icon**.
- Click on **Accept button**.

Wiki extension

Wiki helps you to quickly creating and editing any number of interlinked wiki pages and OpenKM nodes (documents, folders, mails and records). Multiple users can collaborate on it, creating, updating or deleting wiki information.



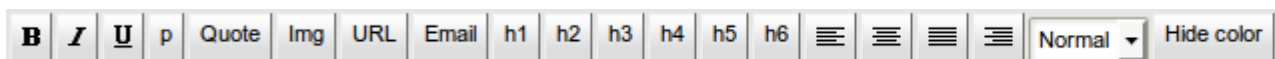
This option is only available if you have enabled the **Wiki** feature.




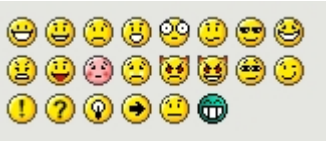







You can set a wiki page for each node.

In Dashboard view, there's a general purpose wiki.

Editor features



Feature	Description
B	<i>Bold.</i>
I	<i>Italic.</i>
U	<i>Underline.</i>
p	<i>New paragraph.</i>
Quote	<i>Make a quotation.</i>
Img	<i>Add image url.</i>
Email	<i>Add email.</i>
h1 to h6	<i>Header settings.</i>
Align Left, center, justify and right.	<i>Set text justification.</i>

Text size	Set text size
Show colour	Palette of colors. 
Smilies	Smilie icons. 
	Add document link.
	Add image.
	Add folder link.
	Add mail link.
	Add record link.
	Add new wiki page.
	Add new wiki link.

Creating a new wiki page

- **Choose a node** (document, folder, mail or record).
- Click on **Create button**.
- **Fill the form**.
- Click on **Create button**.

Update wiki page

- **Choose a node** (document, folder, mail or record).
- Click on **Update button**.
- **Modify the form**.
- Click on **Update button**.

Delete wiki page

- **Choose a node** (document, folder, mail or record).
- Click on **Delete button**.
- Click on **Accept button**.

Show history

- **Choose a node** (document, folder, mail or record).
- Click on **History button**.



From history view you can restore or view previous versions.

Unlock wiki page



Wiki pages are updated by one user at the time. While a page is being updated by an user, nobody else can change it.

When the user click on "Cancel" or "Update" button the wiki page is unlocked.

- Click on **Unlock button**.

Stapling extension



Stapling help you to set relations between documents.



*This option is only available if you have enabled the **Stapling** feature.*


The list of related nodes is based on user security.

Create new stapling group

- **Choose a node** (document, folder or mail).
- Click on  **Add node icon**. At status panel will be shown the message "Stapling started".
- **Choose other node**.
- Click on  **Add node icon**.




At this point you have set a relation between two nodes but you can continue setting relations with the iteration:

- **Choose other node**.
- Click on  **Add node icon**.

- Click on  **Stop stapling icon**.


Working with stapling group

Add new relations

- Click on **Add button**.
- **Choose other node**.
- Click on  **Add node icon**.



At this point you have set a relation between two nodes but you can continue setting relations with the iteration:

- **Choose other node**.
- Click on  **Add node icon**.

- Click on  **Stop stapling icon**.

Delete relations

- Click on  **Delete button**.

Download nodes

- Click on **Download button**.



It downloads ZIP file with folders and document nodes.

Using OpenKM from external applications

There are a number of options available where you can access and use OpenKM from external applications.

Microsoft Office Add-in

You can create, edit and save files that are stored in the OpenKM repository, directly from Microsoft Office applications such as Microsoft Word, Microsoft Excel, Microsoft Power Point and Microsoft Outlook.

With OpenKM Microsoft Outlook Addin you can save mails to OpenKM repository from within Microsoft Outlook.



More information at [Microsoft Office Add-in](#).

Digital Signature Client

Digital Signature Clients help you to quickly sign - electronic digital signature - documents to OpenKM.



More information at [Digital signature client](#).

Scanner client

Scanner Client allows to capture images from scanners and upload them to the OpenKM repository.



Optionally:

- The images can be convert to PDF file.
- Can add metadata group while uploading files to the OpenKM repository.

More information at [Scanner client](#).

Desktop Importer tool

The Desktop Importer tool is designed to import documents to a remote OpenKM instance. It uses OpenKM WebServices API so is firewall friendly.



More information at [Desktop importer tool](#).

Microsoft Office Add-in

You can create, edit and save files that are stored in the OpenKM repository, directly from Microsoft Office applications such as Microsoft Word, Microsoft Excel, Microsoft Power Point and Microsoft Outlook.

With OpenKM Microsoft Outlook Add in you can save mails to OpenKM repository from within Microsoft Outlook.



Importing mail creates a folder structure based on received email date (**year/month/day** folder structure) and imports by default to **"okm:mail/{userId}"** node.

When a user edits a document, they are saved locally in **"%USER_LOCAL_DOCUMENT_FOLDER%/OpenKM"**.

When editing OpenKM documents, a user can decide to upload or cancel editing. After both operations the local file is removed so as not to generate user confusion. When you upload a new file, it is not removed from your computer.

In the file **"%USER_LOCAL_DOCUMENT_FOLDER%/OpenKM/openkm.xml"** are saved the connection OpenKM data. The password is crypted.

In the file **"%USER_LOCAL_DOCUMENT_FOLDER%/OpenKM/document.xml"** are saved the documents edition information status.

There is a log file for each application at:

- **"%USER_LOCAL_DOCUMENT_FOLDER%/OpenKM/WordAddin.log"**.
- **"%USER_LOCAL_DOCUMENT_FOLDER%/OpenKM/ExcelAddin.log"**.
- **"%USER_LOCAL_DOCUMENT_FOLDER%/OpenKM/PowerPointAddin.log"**.
- **"%USER_LOCAL_DOCUMENT_FOLDER%/OpenKM/OutlookAddin.log"**.



Microsoft Office Add in is **compatible with versions 2000 to 2013**.








Configuring add-in

- Click on configuration toolbar icon (menu option at Microsoft Outlook).
- Set version to **"Webservices 2.0"**.
- Choose language (by default English).
- Set the **username**.
- Set the **password**.
- Set the **host** (for example <http://localhost:8080/OpenKM>).
- Click on **Test** button to check the connection configuration.

Editing documents from Microsoft Office



Depending your Microsoft Office version you will see an OpenKM **toolbar** or a **Ribbon**.

Feature	Description
 Configuration	Configure the connection to OpenKM server.
 Add document	Upload a new document into the OpenKM repository.
 Edit	Edit document (check-out). <div>  <p>When a user edits a document, they are saved locally in "%USER_LOCAL_DOCUMENT_FOLDER%/OpenKM".</p> <p>You can continue editing the document other day (the document status is saved at "%USER_LOCAL_DOCUMENT_FOLDER%/OpenKM/document.xml" file.</p> </div>
 Update	Upload a new document version (check-in).
 Cancel edit	Cancel document edition.
 Help	OpenKM to OpenKM Microsoft Office Add in URL in your default web browser.

Importing mails from Microsoft Outlook

- **Choose** one or several **mails**.
- **Click on import** menu option.
- **Follow the wizard** to select the destination folder. When grouped checkbox is enabled will be created the **year/month/day** folder structure.
- Click on **Upload** button.

Add new translation



The translation file is set at "%USER_LOCAL_DOCUMENT_FOLDER%/OpenKM/OKMAddin-enGB.properties". The easiest way is to change the contents of the file directly.

- Create a new translation file at "%USER_LOCAL_DOCUMENT_FOLDER%/OpenKM/" (for example copy OKMAddin-enGB.properties to OKMAddin-esES.properties for Spanish translation).
- Set the translations.
- Edit "%USER_LOCAL_DOCUMENT_FOLDER%/OpenKM/openkm.xml" and change the tag

```
<idioma>enGB</idioma>
```

to

```
<idioma>esES</idioma>
```

Know issues

In Microsoft Office 2010 we have detected some issue, if you detect some problem using this version try it:

- Open the menu **File > Options**.
- Select **Trust Center** and click on "**Trust Center Settings**" button.
- Open **Protected View** option and **disable the first three** checkbox.

Digital Signature Client

Digital Signature Clients help you to quickly sign - electronic digital signature - documents to OpenKM.



In the file "%USER_LOCAL_DOCUMENT_FOLDER%/OpenKM/openkm.xml" are saved the connection OpenKM data. The password is crypted.

The Digital Signature Client takes advantage of using **Microsoft Digital Certificates stores**.

There is a log file for each application at:

- "%USER_LOCAL_DOCUMENT_FOLDER%/OpenKM/DigitalSignature.log".

Configuring

- Click on **File > Configuration**.
- Set version to "**Webservices 2.0**".
- Choose language (by default English).
- Set the **username**.
- Set the **password**.
- Set the **host** (for example <http://localhost:8080/OpenKM>).
- Click on **Test** button to check the connection configuration.

Sign documents

- To sign documents, **select files or folders** with files into and **drag to Digital Signature Client**.
- Will be shown a Pop-up to choose the certificate to be used to sign documents. **Select a certificate** and click the **Accept** button.
- **Select the destination folder** to save the documents and click **Sign** button.



If Digital Signature Client detects an existing document with the same name in the repository, it will show a warning asking to update or skip the document.







Scanner client

Scanner Client allows to capture images from scanners and upload to the OpenKM repository.



Optionally:

- The images can be converted to PDF file.
- Can add metadata group while uploading files to the OpenKM repository.

	Feature	Description
	Scan	Starts scanning process.
	Convert to PDF	Convert an image or group of selected images to a PDF document.
	Refresh	Refreshes tree.
	Upload	Uploads a document.
	Delete PDF page.	Deletes a page from the selected PDF document.
	Add page to PDF	Adds new page to the selected PDF document.

Requirements

Before running the application make sure you have installed:

- Visual C++ Redistributable for Visual Studio
 - .NET 4.0: Visual C++ Redistributable for Visual Studio 2012 ([x86 or x64](#)) (Recommended).
 - .NET 2.0: Visual C++ Redistributable for Visual Studio 2008 ([x86](#) or [x64](#)).
- PDF Acrobat Reader.
- Microsoft .Net Framework 3.5 or upper.

Configuring

- Click on **File > Configuration**.
- Set version to "**Webservices 2.0**".

- Choose language (by default English).
- Set the **username**.
- Set the **password**.
- Set the **host** (for example <http://localhost:8080/OpenKM>).
- Click on **Test** button to check the connection configuration.

Compression CCITT Group4

- Click on **Configuration > Compression group4** menu option.
- Check the **black and white CCITT Group4 compression** checkbox
- Set the **percentage** (Threshold), by default is set to 80%.
- Click on **Accept** button.

Setting destination folder

- Click top right **Change** button.
- Select the destination folder and click on **Select** button.

Setting scanner configuration

- Click on Configuration / **Scanner configuration**.
- Change the configuration parameters.
- Click on **Accept** button.



We suggest TWAIN protocol rather WIA for better scanner features support.

Scanning

- Click on **Scanning** button.
- Follow the wizard.



Will be created a folder based on date (**year-month-day**) into your server at **My images/OKMScan/Images**.

Convert scanned images to PDF

- Choose one or several image files from tree.
- Click on **Images to PDF** button.



A PDF file will be created into a folder based on date (**year-month-day**) into your server at **My images/OKMScan/Pdfs**.


Uploading

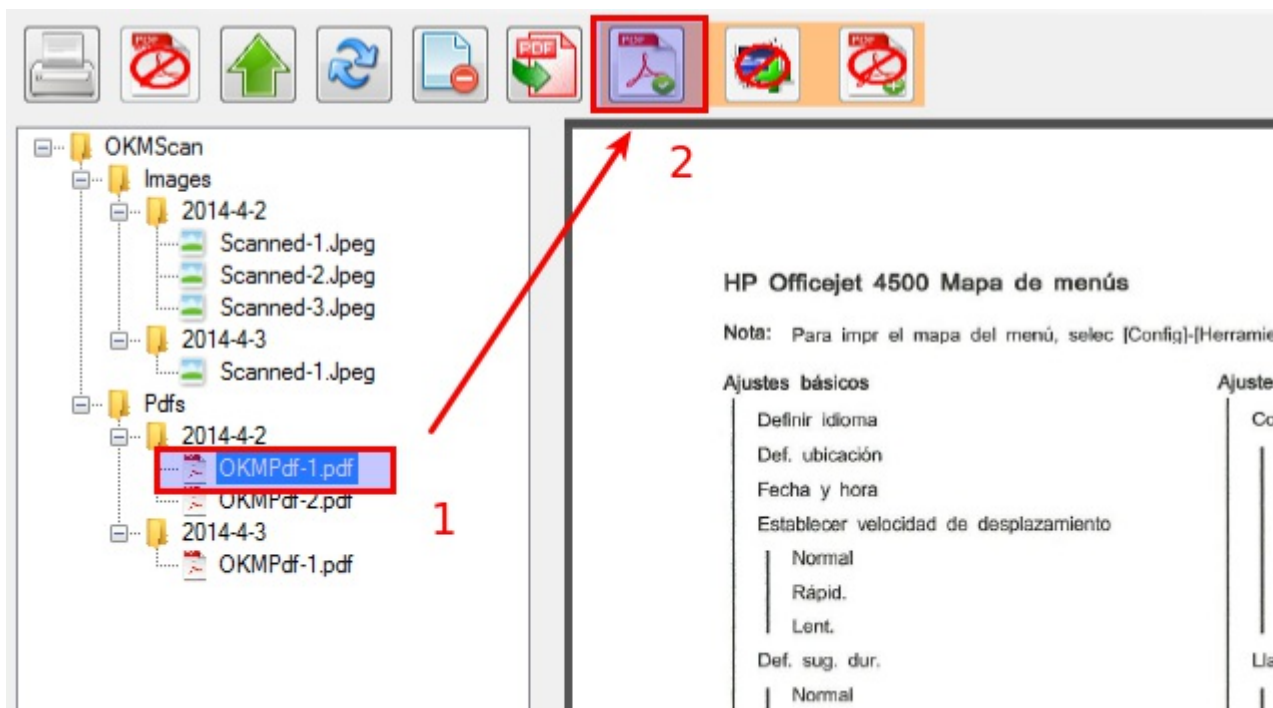
- **Choose one or several files** (images or PDF).
- Click on **Upload** button.
- **Optionally you can set metadata groups.**
 - **Choose the metadata group.**
 - Click on **Add** button.
 - **Fill the metadata group fields.**
 - Click on **Add** button.


Add images to an existing PDF file

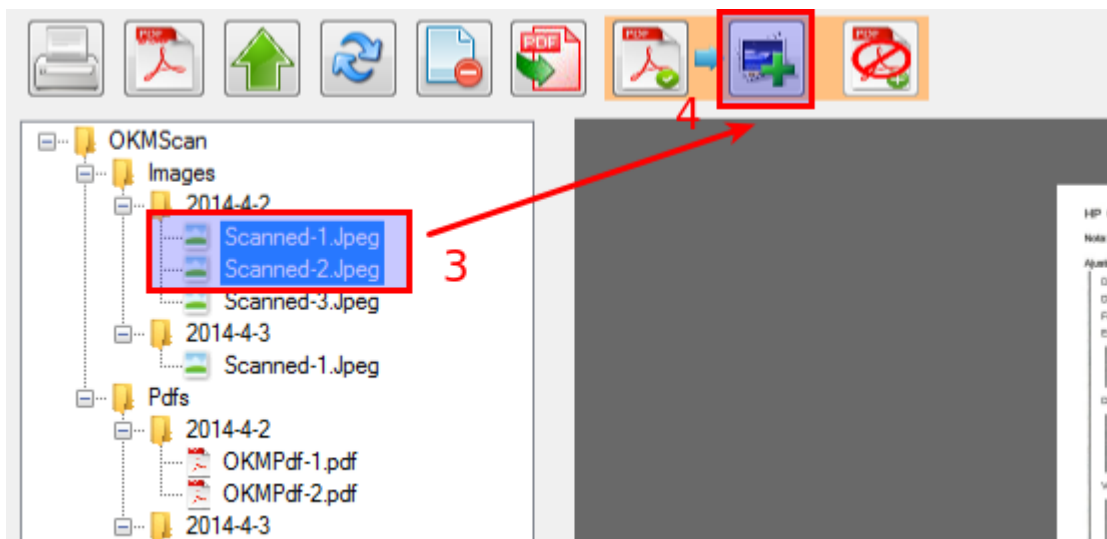
- Click on **Add page to PDF** button. Will be shown extra icons in the toolbar.




- Action has tree steps, first is to **select a PDF file** and then click on the icon  to **confirm PDF selection**.



- The second step is to **select one or several images** and click on the icon  to **confirm images selection**.



- And finally click on the icon  to **add the images**. Will be shown a popup to confirm the action, click on **Yes** button.

Remove pages from existing PDF file

- **Choose a PDF file.**
- Click on **Delete PDF page** button.
- **Select a range** of pages.
- Click on **Accept** button.

Desktop importer tool

This utility is designed to import documents to a remote OpenKM instance. It uses OpenKM WebServices API so is firewall friendly.

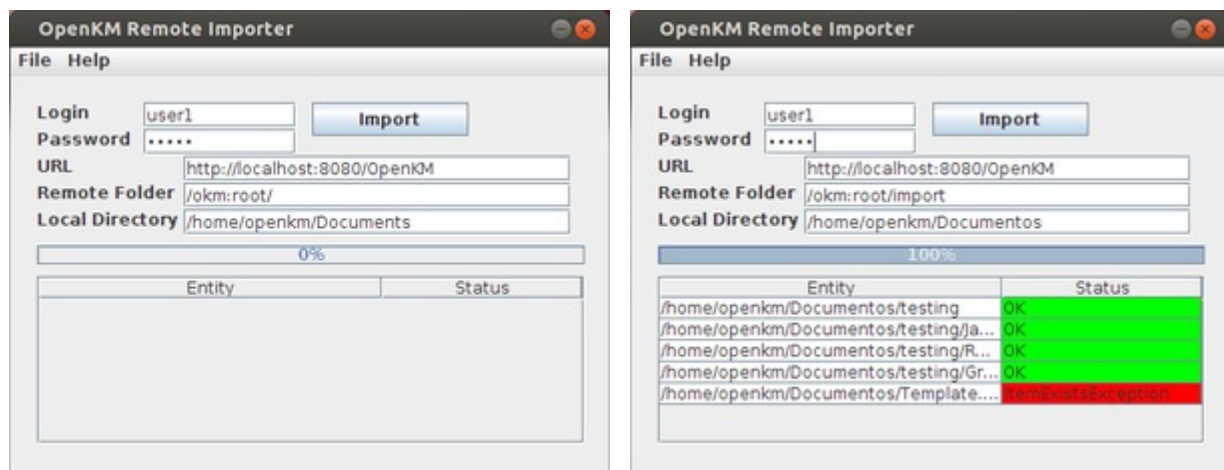


You need Java 7 or upper installed in your computer.



You can directly download and execute the [desktop-importer.jnlp](#) file from the web.

Images



Migration guide

Major OpenKM version implies changes in repository structure. This means that a migration process should be run to make these changes. Sometimes the database structure also is modified and need to be adapted.

Here you can find the migration process between different OpenKM releases:

- [Migrating from 6.3.8 to 6.3.9](#)
- [Migrating from 6.3.7 to 6.3.8](#)
- [Migrating from 6.3.6 to 6.3.7](#)
- [Migrating from 6.3.4 to 6.3.5](#)
- [Migrating from 6.3.3 to 6.3.4](#)
- [Migrating from 6.3.2 to 6.3.3](#)
- [Migrating from 6.3.1 to 6.3.2](#)
- [Migrating from 6.3.0 to 6.3.1](#)
- [Migrating from 6.2.6 to 6.3.0](#)
- [How to switch between databases](#)

Changelog

OpenKM 6.3 Community

v6.3.9

- *Added HTML5 drag & drop feature*
- *Added detailed information in Lucene index process*
- *Added createMissingFolders method in the API*
- *Fixed error in administration when choosing path with special characters*
- *Fixed temp files were not removed*
- *Added isCallingMe method in StacktraceUtils to detect recursive plugin calls*
- *Fixed issue with the logo which was not changing in the login screen*
- *Fixed UI defect with workflow instance with run_config*
- *Fixed maven dependency conflicts*

v6.3.8

- *Fixed CSV exporter error*
- *Fixed popup UI defect in wiki*
- *Prevent from export into \$TOMCAT_HOME/webapps*
- *Added login and logout automation events*
- *Added profile to create JAR client and dependencies*

v6.3.7

- *Fixed error in mobile web interface*
- *Fixed some security vulnerabilities*
- *Fixed minor defect in export and import features*
- *Fixed error when purging trash*
- *Fixed problem with folder notes*
- *Improvements in administration look and feel*
- *Added support for Lucene plain queries*
- *Improved automation using plugins*

v6.3.6

- *Fixed error while setting metadata.*

v6.3.5 - Nexus

- *Fixed problem with automation in document move*
- *Upload now supports files bigger than 2GB*
- *Implemented HTML5 document preview*
- *Added missing search method for compatibility*
- *Several mail related improvements and fixes*
- *OMR feature added*

v6.3.4 - Tachikoma

- *Added a news ticker in login screen with OpenKM latest news*
- *Solve preview issue which requires license key*

v6.3.3 - Reborn

- *Added Oracle support*
- *Updated styles for better look & feel*
- *Changed login screen*
- *Added cache L2 support*
- *Added profiling support*
- *Added hibernate stats support*
- *Added UI Desktop button for splitter resizing*
- *Added extra columns based in metadata in filebrowser view*
- *Added Merge PDF feature*
- *Added Convert feature*
- *Added Stapling extension*
- *Added Dashboard Messaging extension*
- *Added Activity Log extension*
- *Added Forum extension*
- *Added Wiki extension*
- *Added Workflow document info*
- *Added HTML Editor 3 extension*
- *Added Forward E-mail feature*
- *Added Send Document Attachment feature*
- *Removed Dropbox feature. DropBox changes caused stop working and OpenKM team staff decided stop the support*
- *Removed OpenMeetings feature. A lot of API changes between version 2.x and 3.x caused stop working, for same reason we have decided stop the support*

v6.3.2 - Relative

- *Minor changes*

v6.3.1 - Pumpkin

- *Fix error when mapping SuggestBox in workflow forms*
- *Automation "Revoke all users" not working when preserverCreator is false*
- *TextExtraction hangs*
- *Loop in PngTextExtractor in Apache JackRabbit*
- *Print button is not showing right title*
- *A checked-out document shouldn't be unlocked*
- *Show better information in file browser table about from mail account. Now take in consideration more cases*
- *Bug on changing security from UI. An specific security modification steps can cause wrong security changes or UI error*
- *Problem with Unicode character into XML REST response*
- *Restore with UUIDs not working on import*
- *Error when modifying default roles in applicationContext.xml*
- *Be able to execute SQL, HQL and BeanShell from WebServices*
- *Invalid token are not detected in SOAP webservices*
- *Change quirks mode to doctype (solve some IE problems)*
- *Improvements in repository export*
- *Improve PDF text extraction*
- *Set Tesseract as default OCR engine*
- *Correct several Cross Site Scripting vulnerabilities*
- *When login WS are not checking if /okm:trash exists*
- *Some methods fails in REST on version 6.2*
- *Downloaded Greek docs with wrong name when using IE*
- *Improve repository export & import logging*
- *Solve problem with webdav URL and character &*
- *REST serialization problem with Document object*
- *Validators are empty from rest service*
- *Select options now can be loaded from classes*
- *Improve pre-visualization auto-configuration*
- *Improved simple search*
- *Fixed problem with Turkish language*
- *Add trailing "/" when upgrade to Tomcat 8*
- *Fixed problem with property group multiple*

- *Fixed problem when importing mail with very long subject*
- *Some mail server can't sent an email if the from field does not match the authenticated user*

v6.3.0 - *Serpico*

- *Metadata browsing*
- *Document listing paginated and filtered (by type, name, etc.)*
- *Different splitter resize schemas*
- *Choose default start workspace*
- *Massive actions when adding metadata, keywords and categories*
- *Better Chrome and IExplorer support*
- *Released extensions: Workflow properties, OpenMeetings and Zoho integration*
- *Forward imported mail*
- *Improved bulk uploader*
- *Document conversion*
- *Merge PDF documents*
- *Able to select default file browser width*
- *Improved Windows service installation*
- *Increase major / minor document version*
- *Change login page logo*
- *Improved mail importer*
- *CMIS support*
- *New RESTful web services API (v1.0)*
- *New SOAP web services (v1.0)*
- *New Automation actions and events (document update, mail creation, etc.)*
- *Lots of bug fixes and several performance improvements*

Tomcat update

By default OpenKM is installed with Tomcat 7.0.61 but every few weeks a new version is released and maybe you want your installation up to date. In this case you ask in the forum at <http://forum.openkm.com> to ensure this new version has been packaged by the OpenKM Team. Once you have verified and downloaded the new version is available you have to follow these steps:



Before updating Tomcat we recommend to upgrade your OpenKM version to 6.3.3 and ensure you have a recent working backup.



OpenKM versions previous to 6.3.3 are not compatible with Tomcat 7.0.61 , so upgrade your OpenKM instance to this version and after that update Tomcat to 7.0.61.

- Stop the current Tomcat
- Unzip the new downloaded one
- Copy these files and folders from the old Tomcat to the new one:
 - conf/server.xml
 - conf/context.xml
 - OpenKM.cfg
 - OpenKM.xml
 - repository
 - plugins
- Edit the conf/server.xml file and comment or remove these lines:

```
<!-- Automatically reload log4j configuration -->  
<Listener className="com.openkm.util.Log4JInitializer" configFile="${catalina.base}/c
```

- Update your Tomcat service configuration:
 - In case of Linux, change the Tomcat path at **/etc/init.d/tomcat**
 - In case of Windows, change the Tomcat path in the service configuration
- Start the new Tomcat



Verify folder and files permissions after copying or moving them. In Linux you can set them recursively:

```
$ sudo chown openkm:openkm tomcat-7.0.61 -R
```



Remember to update the backup script because the location of the Tomcat directory would change. In case of Linux you can check it by doing (as root):

```
$ crontab -l
```


This way, you can see the location of the backup script.

Preserving the old Tomcat is a good idea, but keep on mind that the backup script usually backups all the /home/openkm directory so move it to another place.

If you want a more recent branch you can try Tomcat 8.5, but keep on mind that this new branch has not been fully tested. In this case further modifications are needed because this new Tomcat uses another pool:

Tomcat 7	Tomcat 8
maxActive	maxTotal
maxWait	maxWaitMillis

Tomcat 7

```
<Resource name="jdbc/OpenKMDS" auth="Container" type="javax.sql.DataSource"
maxActive="100" maxIdle="30" maxWait="10000" validationQuery="select 1"
username="openkm" password="*****" driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost:3306/okmdb?autoReconnect=true&useUnicode=true&characterEncoding=utf8"/>
```

Tomcat 8

```
<Resource name="jdbc/OpenKMDS" auth="Container" type="javax.sql.DataSource"
maxTotal="100" maxIdle="30" maxWaitMillis="10000" validationQuery="select 1"
username="openkm" password="*****" driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost:3306/okmdb?autoReconnect=true&useUnicode=true&characterEncoding=utf8"/>
```

How to switch between databases

The process of migration between two databases can be divided into several steps:

Generate users and roles SQL scripts


From HSQLDB

```
SELECT 'INSERT INTO OKM_USER (USR_ID, USR_NAME, USR_PASSWORD, USR_EMAIL, USR_ACTIVE)
SELECT 'INSERT INTO OKM_ROLE (ROL_ID, ROL_ACTIVE) VALUES ('' + ROL_ID + '', '' + R
SELECT 'INSERT INTO OKM_USER_ROLE (UR_USER, UR_ROLE) VALUES ('' + UR_USER + '', ''
```


From MySQL

```
SELECT CONCAT('INSERT INTO OKM_USER (USR_ID, USR_NAME, USR_PASSWORD, USR_EMAIL, USR_A
SELECT CONCAT('INSERT INTO OKM_ROLE (ROL_ID, ROL_ACTIVE) VALUES ('', ROL_ID, '', '
SELECT CONCAT('INSERT INTO OKM_USER_ROLE (UR_USER, UR_ROLE) VALUES ('', UR_USER, ''
```

Generate database configuration parameters SQL scripts

- Click at top right the **Configuration parameters export icon** .
- Browser will download an SQL file with the configuration.

Export Metadata

- Click top right  **Edit metadata**.
- In Edit view, **copy the XML**.

Export crontab

- Go to **Administration** > [Crontab](#) > **Download**



Download ONLY the tasks that you have created and need to be migrated to the new OpenKM. The other tasks will be automatically created by OpenKM during the installation process.

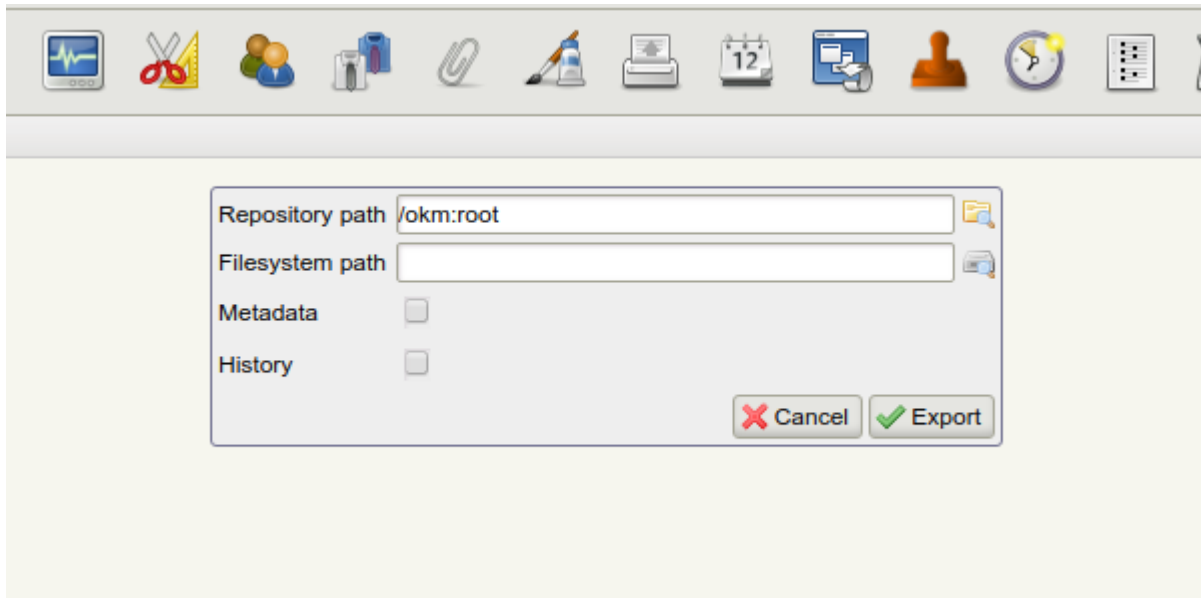
Export the repository



Check if you have enough free space in the file system before exporting the entire repository.

- Go to **Administration** > [Repository export](#)
- Set export repository from node **okm:root**
- Set file system export folder **/mnt/backup/root** (you should create one folder for each principal nodes; /okm:root, /okm:categories, /okm:temples, /okm:personal, /okm:mail, /okm:trash)

- **Check metadata** (mandatory)
- **Check history** (mandatory)
- Click on "**Execute**" button



If you want to export okm:personal among other workspaces you should repeat the process for each one.

Installation of OpenKM



*Install the **same OpenKM version** as you have right now in production.*

Make a new [Installation](#) with the new database server.



From this point, all the steps are done in the new OpenKM installation.

Import the repository

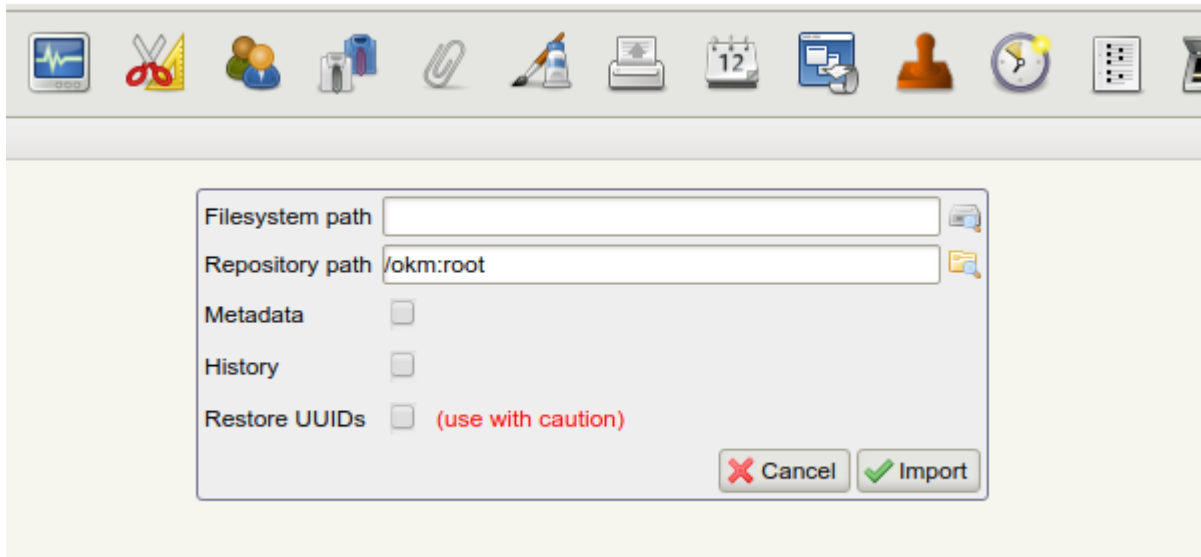


*When you choose the option "**preserve UUIDs**", the nodes UUIDs are preserved while importing. This feature is useful when these UUID's are used by third-party applications and **MANDATORY** when you have categories in your repository.*

Take into consideration that if you are importing into non-empty OpenKM, preserving the UUID's might have some collision with existing ones.

- Go to **Administration** > **Tools** > [Repository export](#)
- Set import repository destination **/okm:root**

- Set file system location of the exported data `/mnt/backup/root`
- Check *metadata and history*
- Check "**Restore UUIDs**"
- Click on "**Execute**" button



Execute scripts


Execute SQL script of users and roles generated at the beginning.

- Go to **Administration** > [Database query](#)
- On the bottom right **choose type JDBC**.
- Write some valid SQL.
- Click on **Execute button**.

Restore configuration parameters



Restoring configuration parameters will remove actual parameters and will be replaced by newer.

- Go to **Administration** > [Configuration parameters](#)
- Go to **bottom page**.
- Select **configuration parameter SQL file**.
- Click  **Import button**.

Restore metadata



To add, modify or delete metadata there's only one operation for all, edit the metadata XML definition.

- Go to **Administration** > [Metadata](#)
- Click top right **Edit metadata**.
- In Edit view, **paste the XML metadata definition** you previously kept safe.
- At bottom right, click **Edit button**.
- At top right, click **Register metadata groups** to apply these changes.
- Add groups to the profile.

Restore crontab task

If you have download Crontab task you might register them again:

- Go to **Administration** > [Crontab](#) and follow the steps to register the tasks as they were in the old OpenKM.

Data what cannot be migrated and might be set again

- [Automation](#).
- [Reports](#).
- [Profiles](#).
- [Activity log](#).



Activity log migration it means migrate all data into **OKM_ACTIVITY** which **might contains millions of rows**. We suggest not to migrate it and only keep safe the old database if you need it in the future.

Migrating from 6.3.9 to 6.3.10



NOT RELEASED

Preliminaries

- *Make a backup!!!*

Migration

- Stop Tomcat
- Edit **OpenKM.cfg** and set **hibernate.hbm2ddl** to **update**
- Replace the OpenKM.war
- And start Tomcat again



If you have any weird problem try to stop Tomcat, delete these folders:

- `$TOMCAT_HOME/webapps/OpenKM`
- `$TOMCAT_HOME/work/Catalina/localhost`



Recommended to clear browser cache and [Java Plugin cache](#).

Migrating from 6.3.8 to 6.3.9

Preliminaries

- *Make a backup!!!*

Migration

- Stop Tomcat
- Edit **OpenKM.cfg** and set **hibernate.hbm2ddl** to **update**
- Replace the OpenKM.war
- And start Tomcat again



If you have any weird problem try to stop Tomcat, delete these folders:

- `$TOMCAT_HOME/webapps/OpenKM`
- `$TOMCAT_HOME/work/Catalina/localhost`



Recommended to clear browser cache and [Java Plugin cache](#).

Migrating from 6.3.7 to 6.3.8

Preliminaries

- *Make a backup!!!*

Migration

- Stop Tomcat
- Edit **OpenKM.cfg** and set **hibernate.hbm2ddl** to **update**
- Replace the OpenKM.war
- And start Tomcat again



If you have any weird problem try to stop Tomcat, delete these folders:

- `$TOMCAT_HOME/webapps/OpenKM`
- `$TOMCAT_HOME/work/Catalina/localhost`



Recommended to clear browser cache and [Java Plugin cache](#).

Migrating from 6.3.6 to 6.3.7

Preliminaries

- Make a backup!!!

Migration

- Stop Tomcat
- Edit **OpenKM.cfg** and set **hibernate.hbm2ddl** to **update**
- Replace the OpenKM.war
- And start Tomcat again

Update automation definition



Execute the scripts below only if you have automation definitions

From version 6.3.7 automation are based on plugins architecture, that means must migrate older database structure to newer. Until the data will be migrated the automation section from the administration will raise errors.

MySQL

Execute the UPDATE queries result of the execution of the next queries:

```
SELECT CONCAT('UPDATE OKM_AUTO_ACTION SET AAC_CLASS_NAME='', m.AMD_CLASS_NAME , ''')
SELECT CONCAT('UPDATE OKM_AUTO_VALIDATION SET AVL_CLASS_NAME='', m.AMD_CLASS_NAME , ''')
```

PostgreSQL

Execute the UPDATE queries result of the execution of the next queries:

```
SELECT 'UPDATE OKM_AUTO_ACTION SET AAC_CLASS_NAME='' || m.AMD_CLASS_NAME || '' WHERE'
SELECT 'UPDATE OKM_AUTO_VALIDATION SET AVL_CLASS_NAME='' || m.AMD_CLASS_NAME || '' WHERE'
```

HQL

Execute the UPDATE queries result of the execution of the next queries:

```
SELECT CONCAT('UPDATE OKM_AUTO_ACTION SET AAC_CLASS_NAME='', m.AMD_CLASS_NAME , ''')
SELECT CONCAT('UPDATE OKM_AUTO_VALIDATION SET AVL_CLASS_NAME='', m.AMD_CLASS_NAME , ''')
```

Custom automation actions or validations

In case you have build your own custom automation actions or validations you should migrate your classes to the new plugin

architecture. Consider reading the documentation sections below for better understanding of the migration process to the new architecture.

Deprecated Automation documentation:

- [Creating your own Automation Action \(deprecated \)](#)
- [Creating your own Automation Validation \(deprecated \)](#)

Current Automation documentation:

- [Creating your own Automation Action](#)
- [Creating your own Automation Validation](#)

All databases



Execute in all the cases.

```
UPDATE OKM_CONFIG SET CFG_TYPE='html' WHERE CFG_KEY='notification.message.body';
UPDATE OKM_CONFIG SET CFG_TYPE='html' WHERE CFG_KEY='proposed.subscription.message.bo
UPDATE OKM_CONFIG SET CFG_TYPE='html' WHERE CFG_KEY='subscription.message.body';
```

Before executing the scripts below, ensure automation data has been migrated correctly, then you can remove it.

```
DROP TABLE OKM_AUTO_METADATA;
ALTER TABLE OKM_AUTO_ACTION DROP COLUMN AAC_TYPE;
ALTER TABLE OKM_AUTO_VALIDATION DROP COLUMN AVL_TYPE;
```



If you have any weird problem try to stop Tomcat, delete these folders:

- \$TOMCAT_HOME/webapps/OpenKM
- \$TOMCAT_HOME/work/Catalina/localhost



Recommended to clear browser cache and [Java Plugin cache](#).

Migrating from 6.3.5 to 6.3.6

Preliminars

- *Make a backup!!!*

Migration

- Stop Tomcat
- Edit **OpenKM.cfg** and set **hibernate.hbm2ddl** to **update**
- Replace the OpenKM.war
- And start Tomcat again



If you have any weird problem try to stop Tomcat, delete these folders:

- `$TOMCAT_HOME/webapps/OpenKM`
- `$TOMCAT_HOME/work/Catalina/localhost`



Recommended to clear browser cache and [Java Plugin cache](#).

Migrating from 6.3.4 to 6.3.5

Preliminars

- Make a backup!!!

Migration

- Stop Tomcat
- Edit **OpenKM.cfg** and set **hibernate.hbm2ddl** to **update**
- Replace the OpenKM.war
- And start Tomcat again
- Go to Administration > [Scripting](#) execute

```
import com.openkm.core.DatabaseException;
import com.openkm.dao.bean.CronTab;
import com.openkm.dao.CronTabDAO;

static final String MAIL = "noreply@openkm.com";

try {
    for (CronTab ct : CronTabDAO.findAll()) {
        if (ct.getMail().equals(MAIL)) {
            ct.setMail("");
            CronTabDAO.update(ct);
        }
    }
} catch (Exception e) {
    print("Exception: " + e);
}
```



If you have any weird problem try to stop Tomcat, delete these folders:

- \$TOMCAT_HOME/webapps/OpenKM
- \$TOMCAT_HOME/work/Catalina/localhost



Recommended to clear browser cache and [Java Plugin cache](#).

Migrating from 6.3.3 to 6.3.4

Preliminars

- *Make a backup!!!*

Migration

Stop Tomcat

- Edit **OpenKM.cfg** and set **hibernate.hbm2ddl** to **update**.
- Replace the OpenKM.war
- And start Tomcat again.



If you have any weird problem try to stop Tomcat, delete these folders:

- `$TOMCAT_HOME/webapps/OpenKM`
- `$TOMCAT_HOME/work/Catalina/localhost`



Recommended to clear browser cache and [Java Plugin cache](#).

Migrating from 6.3.2 to 6.3.3

Preliminars

- Make a backup!!!

Migration

- Go to **Administration** > [Database query](#) and execute:

HSQL:

```
-- ADD COLUMN NDC_ENCRYPTION
ALTER TABLE OKM_NODE_DOCUMENT ADD COLUMN NDC_ENCRYPTION CHAR(1);
UPDATE OKM_NODE_DOCUMENT SET NDC_ENCRYPTION='T';
ALTER TABLE OKM_NODE_DOCUMENT ALTER COLUMN NDC_ENCRYPTION SET NOT NULL;
-- ADD COLUMN NDC_SIGNED
ALTER TABLE OKM_NODE_DOCUMENT ADD COLUMN NDC_SIGNED CHAR(1);
UPDATE OKM_NODE_DOCUMENT SET NDC_SIGNED='T';
ALTER TABLE OKM_NODE_DOCUMENT ALTER COLUMN NDC_SIGNED SET NOT NULL;
```

MySQL:

```
-- ADD COLUMN NDC_ENCRYPTION
ALTER TABLE OKM_NODE_DOCUMENT ADD COLUMN NDC_ENCRYPTION CHAR(1);
UPDATE OKM_NODE_DOCUMENT SET NDC_ENCRYPTION='F';
ALTER TABLE OKM_NODE_DOCUMENT MODIFY COLUMN NDC_ENCRYPTION CHAR(1) NOT NULL;
-- ADD COLUMN NDC_SIGNED
ALTER TABLE OKM_NODE_DOCUMENT ADD COLUMN NDC_SIGNED CHAR(1);
UPDATE OKM_NODE_DOCUMENT SET NDC_SIGNED='F';
ALTER TABLE OKM_NODE_DOCUMENT MODIFY COLUMN NDC_SIGNED CHAR(1) NOT NULL;
```

PostgreSQL:

```
-- ADD COLUMN NDC_ENCRYPTION
ALTER TABLE OKM_NODE_DOCUMENT ADD COLUMN NDC_ENCRYPTION CHAR(1);
UPDATE OKM_NODE_DOCUMENT SET NDC_ENCRYPTION='F';
ALTER TABLE OKM_NODE_DOCUMENT ALTER COLUMN NDC_ENCRYPTION SET NOT NULL;
-- ADD COLUMN NDC_SIGNED
ALTER TABLE OKM_NODE_DOCUMENT ADD COLUMN NDC_SIGNED CHAR(1);
UPDATE OKM_NODE_DOCUMENT SET NDC_SIGNED='F';
ALTER TABLE OKM_NODE_DOCUMENT ALTER COLUMN NDC_SIGNED SET NOT NULL;
```

- Stop Tomcat
- Edit **OpenKM.cfg** and set **hibernate.hbm2ddl** to **update**.
- Edit **OpenKM.xml** and modify

```
http://www.springframework.org/schema/security/spring-security-3.1.xsd
```

to

```
http://www.springframework.org/schema/security/spring-security-3.2.xsd
```

- *Replace the OpenKM.war*
- *And start Tomcat again.*



If you have any weird problem try to stop Tomcat, delete these folders:

- *\$TOMCAT_HOME/webapps/OpenKM*
- *\$TOMCAT_HOME/work/Catalina/localhost*



Recommended to clear browser cache and [Java Plugin cache](#).

Migrating from 6.3.1 to 6.3.2

Preliminars

- *Make a backup!!!*

Migration

- Stop Tomcat
- Edit OpenKM.cfg and set **hibernate.hbm2ddl** to **update**
- Replace the OpenKM.war
- And start Tomcat again.



If you have any weird problem try to stop Tomcat, delete these folders:

- \$TOMCAT_HOME/webapps/OpenKM
- \$TOMCAT_HOME/work/Catalina/localhost



Recommended to clear browser cache and [Java Plugin cache](#).

Migrating from 6.3.0 to 6.3.1

Preliminars

- Make a backup!!!

Migration

- Stop Tomcat.
- Edit OpenKM.cfg and set **hibernate.hbm2ddl** to **update**.
- Replace the OpenKM.war.
- Execute these queries:

```
DELETE FROM OKM_CONFIG WHERE CFG_KEY='system.ghostscript.ps2pdf';  
UPDATE OKM_CONFIG SET CFG_VALUE='OpenKM - NOTIFICATION' WHERE CFG_KEY='notification.m  
UPDATE OKM_CONFIG SET CFG_VALUE='Message: ${notificationMessage} User: ${userId} <#li
```

- And start Tomcat again.



If you have any weird problem try to stop Tomcat, delete these folders:

- \$TOMCAT_HOME/webapps/OpenKM
- \$TOMCAT_HOME/work/Catalina/localhost



Recommended to clear browser cache and [Java Plugin cache](#).

Migrating from 6.2.6 to 6.3.0

Preliminars

- Make a backup!!!

Migration

- Stop Tomcat.
- Edit OpenKM.cfg file and set **hibernate.hbm2ddl** to **update**.
- Replace the OpenKM.war.
- Go to **Administration** > **Crontab** and delete "User Items Serialize" and "Refresh User Items" tasks.
- Run from your preferred SQL client these queries:

MySQL

```
ALTER TABLE OKM_MIME_TYPE ADD COLUMN MT_SEARCH CHAR(1);
UPDATE OKM_MIME_TYPE SET MT_SEARCH='T';
ALTER TABLE OKM_MIME_TYPE MODIFY COLUMN MT_SEARCH CHAR(1) NOT NULL;
```

PostgreSQL

```
ALTER TABLE OKM_MIME_TYPE ADD COLUMN MT_SEARCH CHAR(1);
UPDATE OKM_MIME_TYPE SET MT_SEARCH='T';
ALTER TABLE OKM_MIME_TYPE ALTER COLUMN MT_SEARCH SET NOT NULL;
```

- Start Tomcat again.
- Go to Administration > Database Query, and execute:

HSQL

```
ALTER TABLE OKM_MIME_TYPE ADD COLUMN MT_SEARCH CHAR(1);
UPDATE OKM_MIME_TYPE SET MT_SEARCH='T';
ALTER TABLE OKM_MIME_TYPE ALTER COLUMN MT_SEARCH SET NOT NULL;
```

- The ExecuteScription action SQL insert has changed between version 6.2.x to 6.3.x please take a look at [Enable Automation actions](#) and [Enable Automation validations](#) see how column with older value **textarea** now must have value **code**.
- Due change in way how some special characters like '&' are stored now in repository is necessary execute this script.
Go to **Administration, Scripting** and execute it.

```
import com.openkm.dao.bean.*;
import com.openkm.dao.*;
```

```
import com.openkm.util.*;
import org.hibernate.*;

String qs = "from NodeBase nb where nb.name like '%&%'" and nb.uuid NOT IN (select nb.
Session session = null;
Transaction tx = null;

try {
    session = HibernateUtil.getSessionFactory().openSession();
    tx = session.beginTransaction();
    Query q = session.createQuery(qs);

    for (NodeBase nBase : q.list()) {
        print(nBase.getUuid() + " -> " + nBase.getName() + "<br>");
        nBase.setName(PathUtils.encodeEntities(nBase.getName()));
        session.update(nBase);
    }

    HibernateUtil.commit(tx);
} catch (Exception e) {
    HibernateUtil.rollback(tx);
    e.printStackTrace();
} finally {
    HibernateUtil.close(session);
}
```

- Enjoy OpenKM 6.3.0!
- And start Tomcat again.



If you have any weird problem try to stop Tomcat, delete these folders:

- \$TOMCAT_HOME/webapps/OpenKM
- \$TOMCAT_HOME/work/Catalina/localhost



Recommended to clear browser cache and [Java Plugin cache](#).

Development

OpenKM 6.3 community portable development environment

The OpenKM Community Portable development environment comes with all in one configured tools.



We encourage using our OpenKM portable development environment rather configure your own Eclipse, specially for Java beginners.

We've tried to provide a development platform as easy as possible to get it running with few steps.

OpenKM development environment comes as a Virtual machine, you can download from <https://www.openkm.com/en/download.html> in this website section you have available a video what try to explain how installing it.



The OpenKM Community portable development environment available from sourceforge is deprecated and should not be used.

Tools:

- Eclipse
- Java 8
- MySQL 5
- MySQL Workbench
- iReport
- Tesseract
- OpenOffice
- Geany
- jBPM 3.3.1.GA
- Firefox

Code:

- Crontab samples.
- Report samples.
- Workflow course.

Suggested minimum hardware configuration:

- 2 CPU's
- 4GB RAM
- 64 bits architecture

Javadoc

For each OpenKM version, it is created a complete Javadoc available at <http://docs.openkm.com/apidoc/>.

Configuring IDE from a clean Eclipse



We recommend using our OpenKM Portable Development IDE rather than configuring your own.

You can also use [IntelliJ IDEA](#) or other development Java tools like [NetBeans](#).

Requirements

Software:

- Eclipse
- OpenJDK 1.8



Your eclipse must have support for:

- Maven
- Subversion (recommended)

Hardware:

Suggested minimum hardware configuration:

- 2 CPU's
- 4GB of ram
- 64 bits architecture (although you can get running on 32 we do not suggest this architecture).

Download eclipse

We suggest downloading the latest stable version of Eclipse. Eclipse comes with several versions, you should download the "Eclipse IDE for Java EE Developers" for your computer architecture (32 or 64 bits).



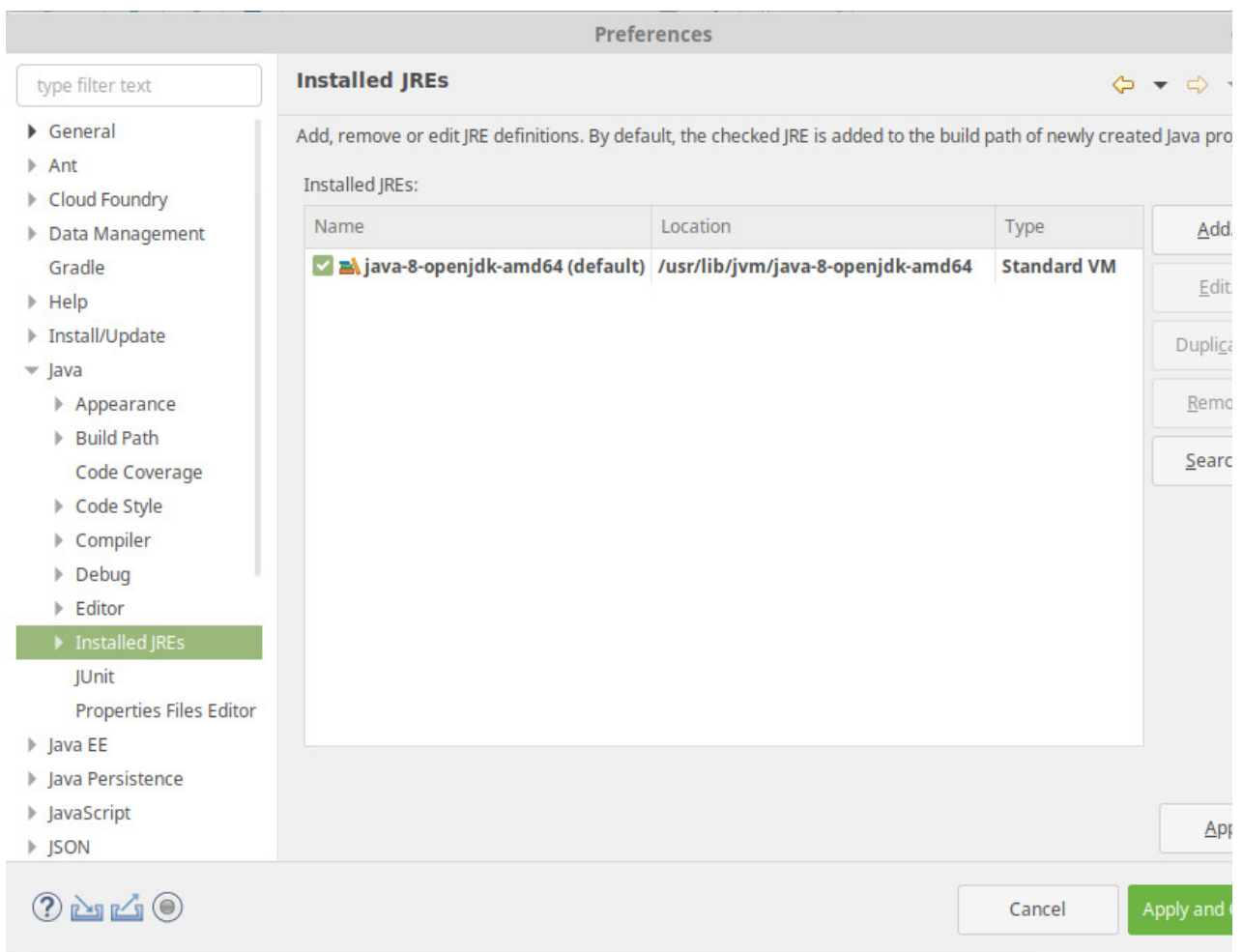
More information at <https://eclipse.org/downloads/>.

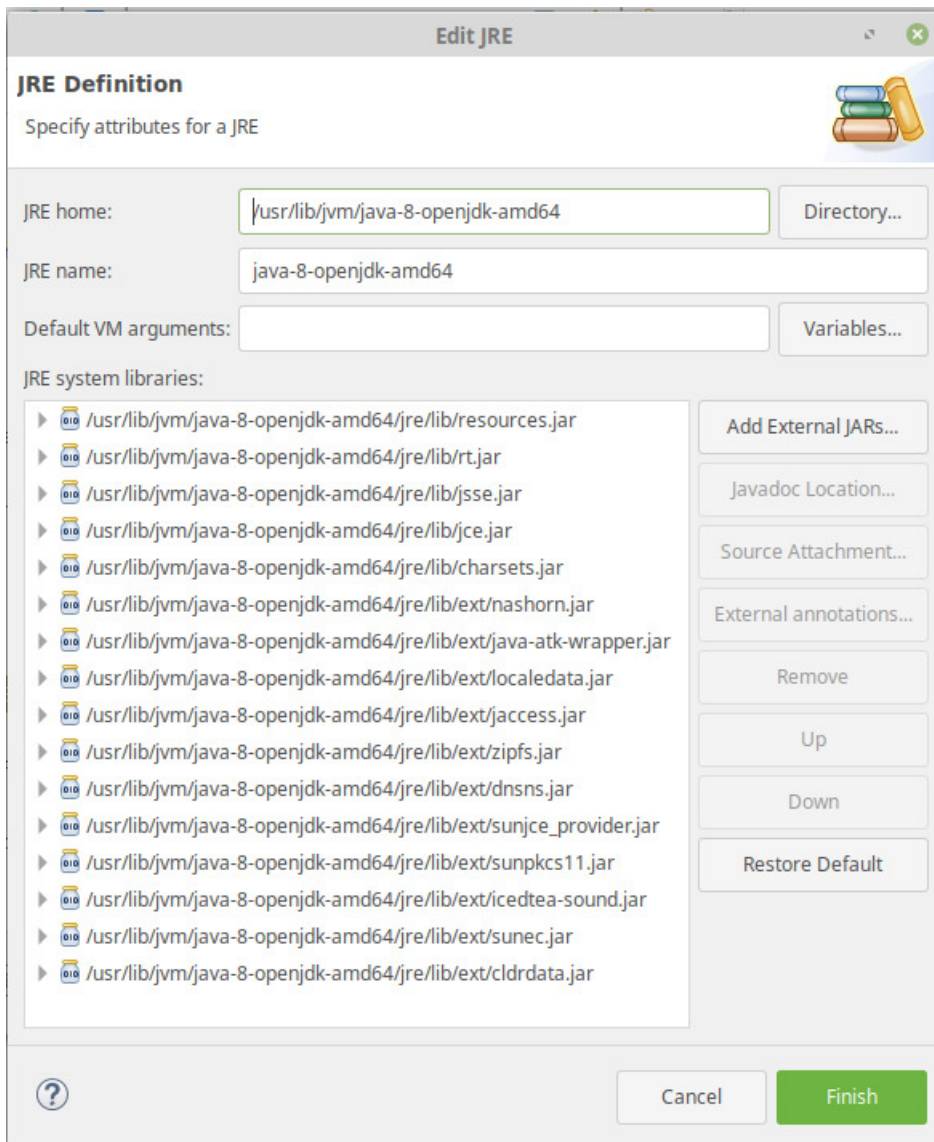
Configure Java



You must have Java installed in your server otherwise go to <http://www.oracle.com/technetwork/java/index.html> click at [Java SE](#) and download the latest OpenJDK 1.8 version according your OS and hardware architecture. Execute the Java installer and follow the installation wizard.

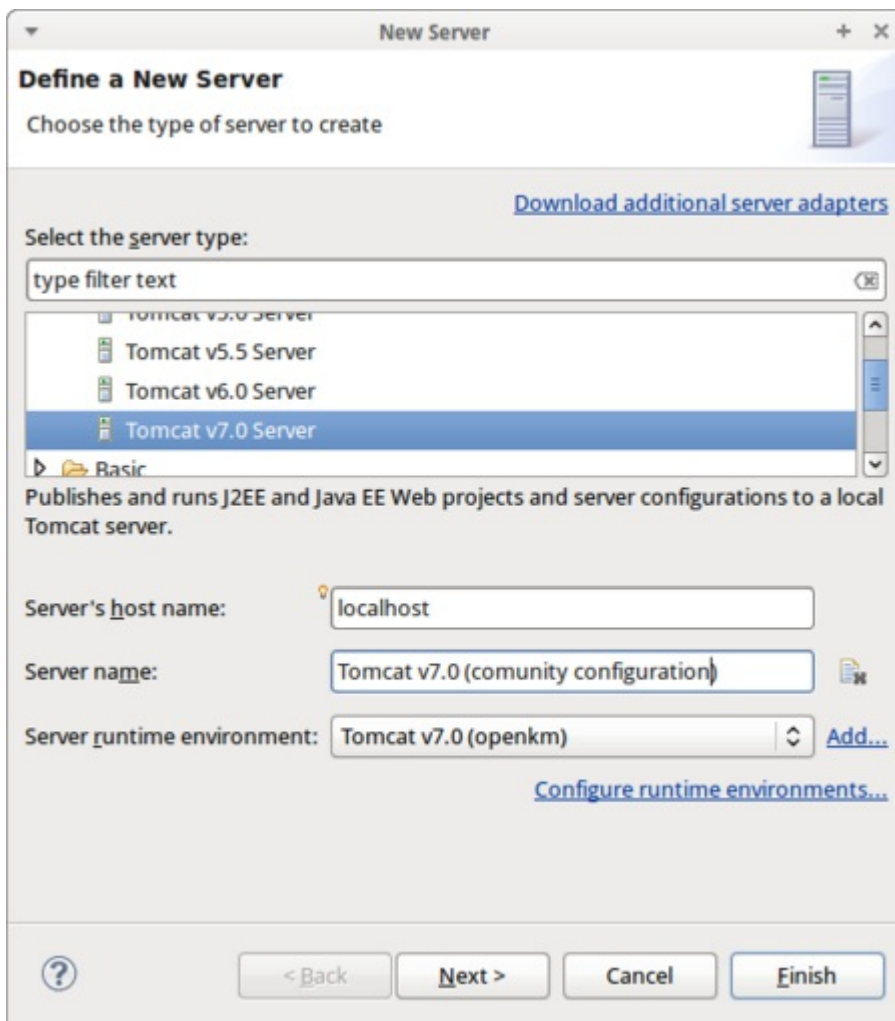
Go to **Window / Preferences / Java / Installed JRE**. You must have OpenJDK 1.8 defined here:





Configure tomcat server

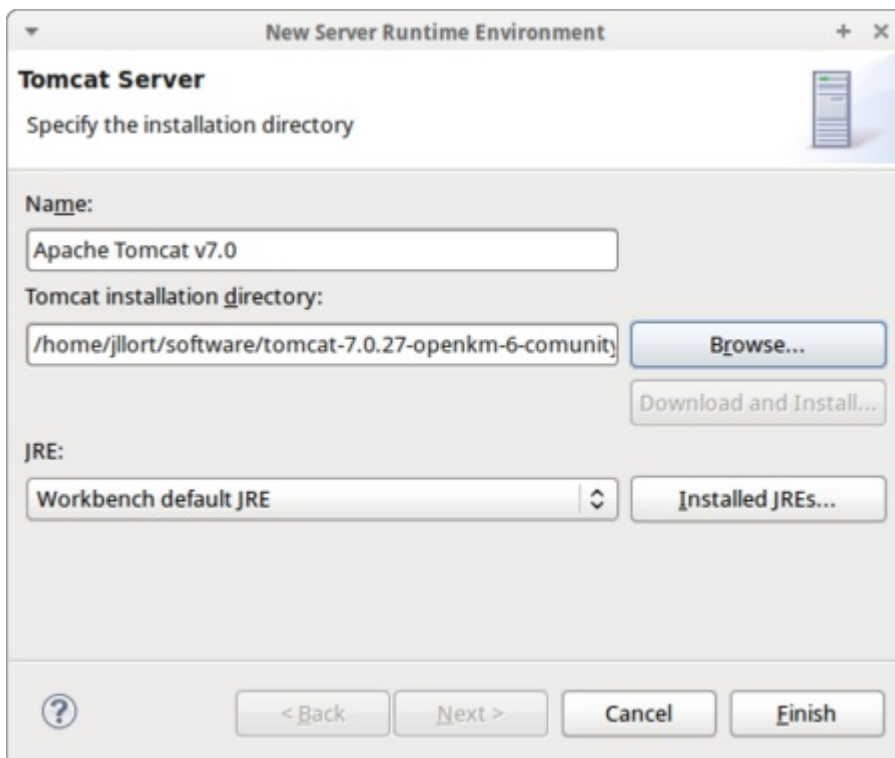
Go to **"servers"** tab. Click on right click mouse button and from contextual menu choose **"Add / server"**. From the list choose **"Tomcat 7 server"**:



Configure a new **Run time server environment**:




Choose the "Workbench default JRE" when your default JRE is "OpenJDK 1.8" or choose "OpenJDK 1.8" from JRE list of installed JREs.



Check box **Use Tomcat installation** (takes control of tomcat installation).

Increment start and stop **timeouts** (for example to 180 and 120 seconds respectively).

Tomcat v7.0 (community configuration) 

Overview

General Information

Specify the host name and other common settings.

Server name:

Host name:

Runtime Environment:

Configuration path:

[Open launch configuration](#)

Server Locations

Specify the server path (i.e. catalina.base) and deploy path. Server must be published with no modules present to make changes.

☒ Use workspace metadata (does not modify Tomcat installation)

☐ Use Tomcat installation (takes control of Tomcat installation)

☐ Use custom location (does not modify Tomcat installation)

Server path:

[Set deploy path to the default value \(currently set\)](#)

Deploy path:

Server Options

Enter settings for the server.

☐ Serve modules without publishing

☐ Publish module contexts to separate XML files

☒ Modules auto reload by default

☐ Enable security

☐ Enable Tomcat debug logging (not supported by this Tomcat version)

Publishing

Timeouts

Specify the time limit to complete server operations.

Start (in seconds):

Stop (in seconds):

Ports

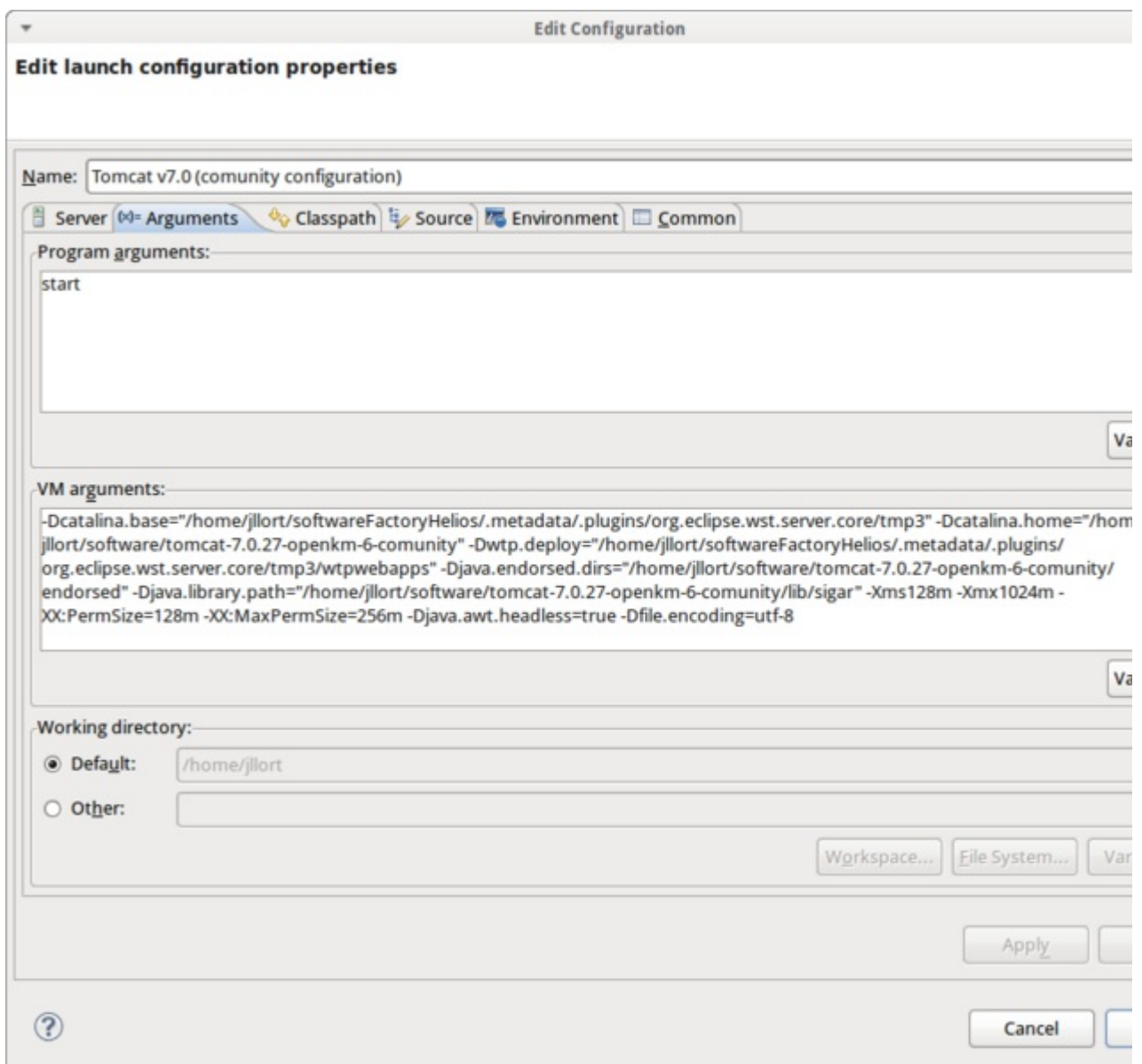
Modify the server ports.

Port Name	P
Tomcat admin port	8
HTTP/1.1	8
AJP/1.3	8

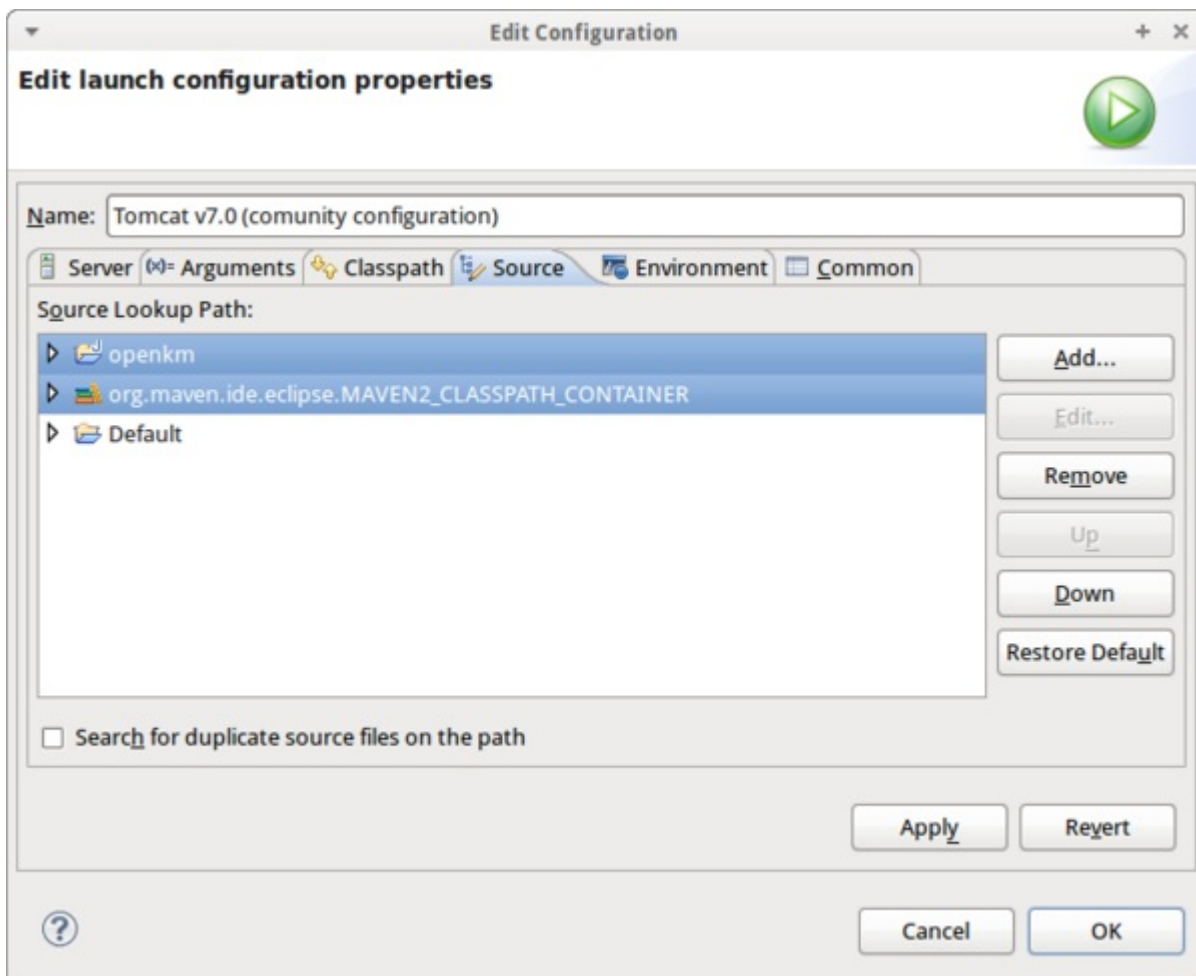
MIME Mappings

*Edit launch configuration: Add **memory parameters** - we suggest 2GB - **sigar** library path and **utf-8** file encoding at VM arguments field:*

```
-Xms128m -Xmx2048m -Djava.awt.headless=true -Dfile.encoding=utf-8 -Djava.library.path
```



Add your **OpenKM project** at source tab:

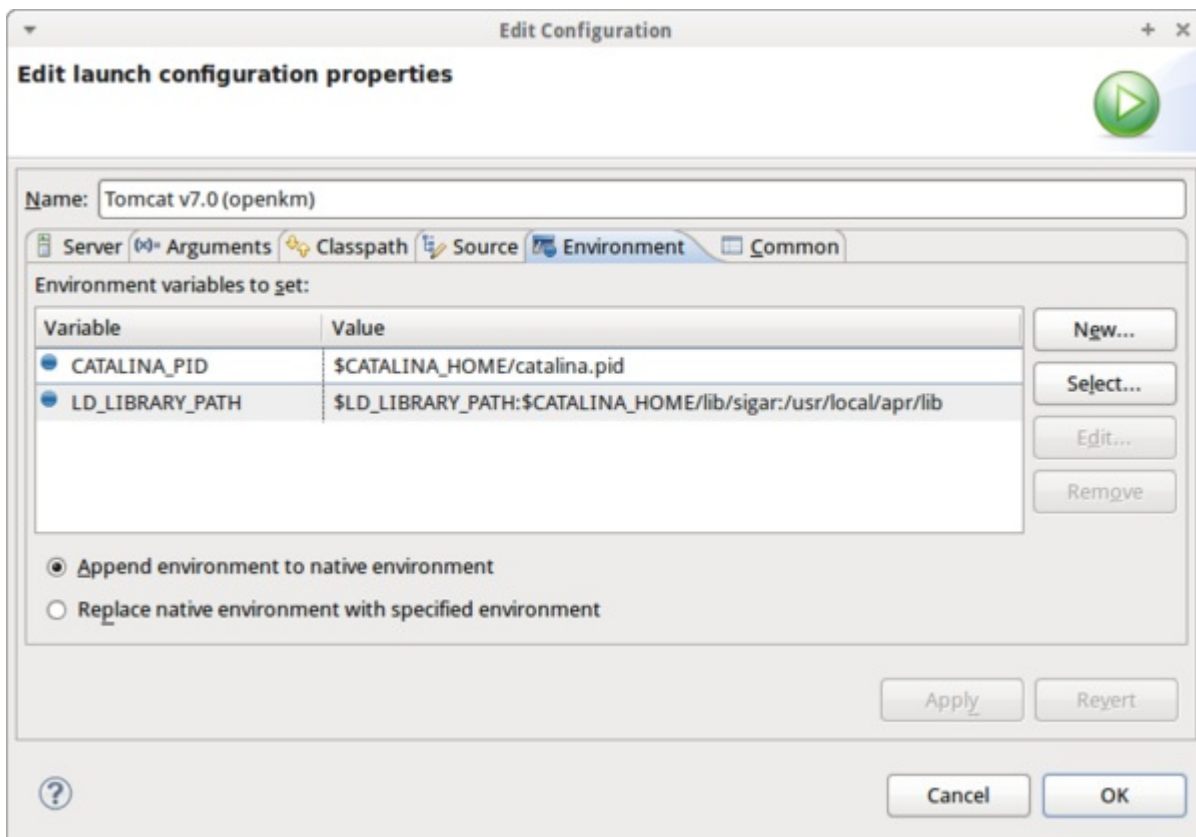


Add **CATALINA_PID** and **LD_LIBRARY_PATH** at environment variables:

```
CATALINA_PID=$CATALINA_HOME/catalina.pid  
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$CATALINA_HOME/lib/sigar:/usr/local/apr/lib
```

or

```
CATALINA_PID=$CATALINA_HOME/catalina.pid  
LD_LIBRARY_PATH=/home/jllort/software/tomcat-7.0.27-openkm-6-community/lib/sigar
```



Maven support for Eclipse

Latest Eclipse version comes by default with maven support plugins yet installed. For older Eclipse version follow this configuration steps:

- Go to **Help > Eclipse Marketplace**
- Install the plugin named [M2Eclipse](#).



Older Eclipse version comes without the "Eclipse Marketplace" feature and the plugins must be installed manually with **Help > New software** option. We encourage do use a more recent version of Eclipse software.

Subversion support for Eclipse

- Go to **Help > Eclipse Marketplace**
- Install the plugin named [Subversive](#).



To checkout a SVN project go to **File > New > Others > Maven > Check out Maven Projects from SCM**.

At the beginning the SCM URL list will be empty, you must click on right bottom "**m2e Marketplace**" link and install "**m2e-subversive**" connector.

Older Eclipse version comes without the "Eclipse Marketplace" feature and the plugins must be installed



*manually with **Help** > **New software** option. We encourage do use a more recent version of Eclipse software.*

Configuring Maven in your OS



The latest Eclipse version comes with embedded Maven software already installed. We suggest using the default configuration.

Maven is a software tool for Java project management and build automation. It is similar in functionality to the Apache Ant tool, but is based on different concepts. Maven is hosted by the Apache Software Foundation, where it was formerly part of the Jakarta Project.

Maven uses a process known as a Project Object Model (POM) to describe the software project being built, its dependencies on other external modules and components, and the build order. It comes with pre-defined targets for performing certain well defined tasks such as compilation of code and its packaging.



More information at <http://maven.apache.org>.



Install Maven version 3.3.x

Follow the [Maven installation guide](#).

Maven tips

Download sources and javadoc from libraries dependency to improve the developer experience:

```
$ mvn install -DdownloadSources=true -DdownloadJavadocs=true
```

Ignore test failures in the build process this way:

```
$ mvn install -Dmaven.test.failure.ignore=true
```

To skip tests, you can do so by executing the following command:

```
$ mvn install -Dmaven.test.skip=true
```

Memory problems

To set more memory for Maven in GNU/Linux you can add the Maven executable to the binary path editing the `$HOME/.bashrc` file and adding this line at the end:

```
export PATH=$PATH:/path/to/maven/installation/bin
export MAVEN_OPTS="-Xmx512m"
```

On windows set from general environment variables feature.

Configure eclipse to use local Maven

Go to **Window > Preferences > Maven > Installation** and click on **Add button** to set your local Maven.



We do not suggest you to apply this change, default embedded Maven configuration usually goes right on almost cases.

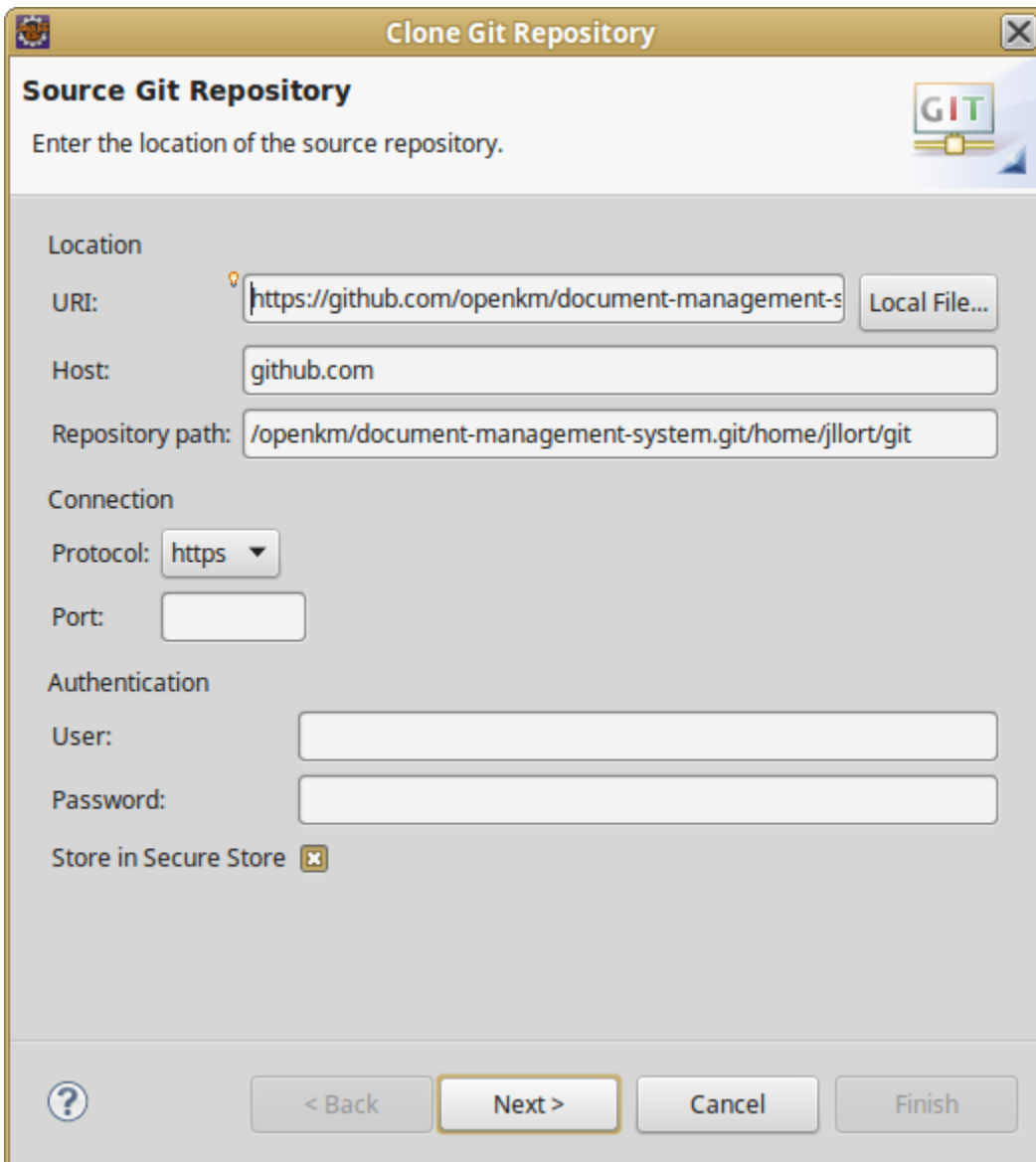
Checkout source code from Git

Clone the Git repository you can do it from eclipse. Go to "**Git Repositories**" and click on "**Git Clone**" icon.



Version	Url	Command line
6.3.x	https://github.com/openkm/document-management-system.git	<pre>\$ git clone https://github.com/openkm/document-management-system.git</pre>

Set the URI:



The image shows a 'Clone Git Repository' dialog box with a yellow title bar and a close button. The main area is titled 'Source Git Repository' and contains the instruction 'Enter the location of the source repository.' with a Git logo. The 'Location' section has three fields: 'URI' with the value 'https://github.com/openkm/document-management-s', 'Host' with 'github.com', and 'Repository path' with '/openkm/document-management-system.git/home/jllort/git'. A 'Local File...' button is next to the URI field. The 'Connection' section has a 'Protocol' dropdown set to 'https' and an empty 'Port' field. The 'Authentication' section has empty 'User' and 'Password' fields. A 'Store in Secure Store' checkbox is checked. At the bottom, there is a help icon, a '< Back' button, a 'Next >' button (highlighted with a yellow border), a 'Cancel' button, and a 'Finish' button.

Clone Git Repository

Source Git Repository

Enter the location of the source repository.

Location

URI:

Host:

Repository path:

Connection

Protocol:

Port:

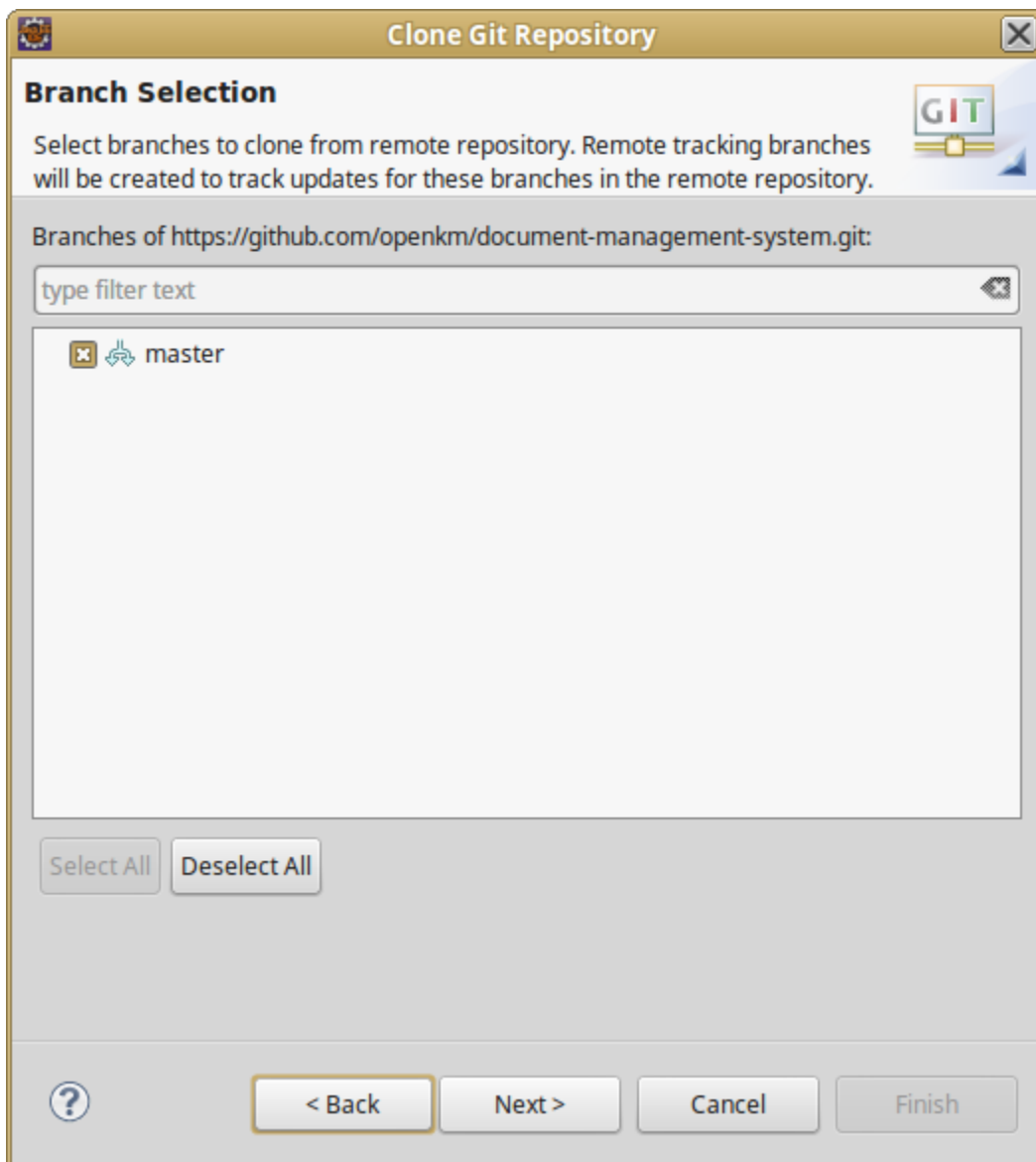
Authentication

User:

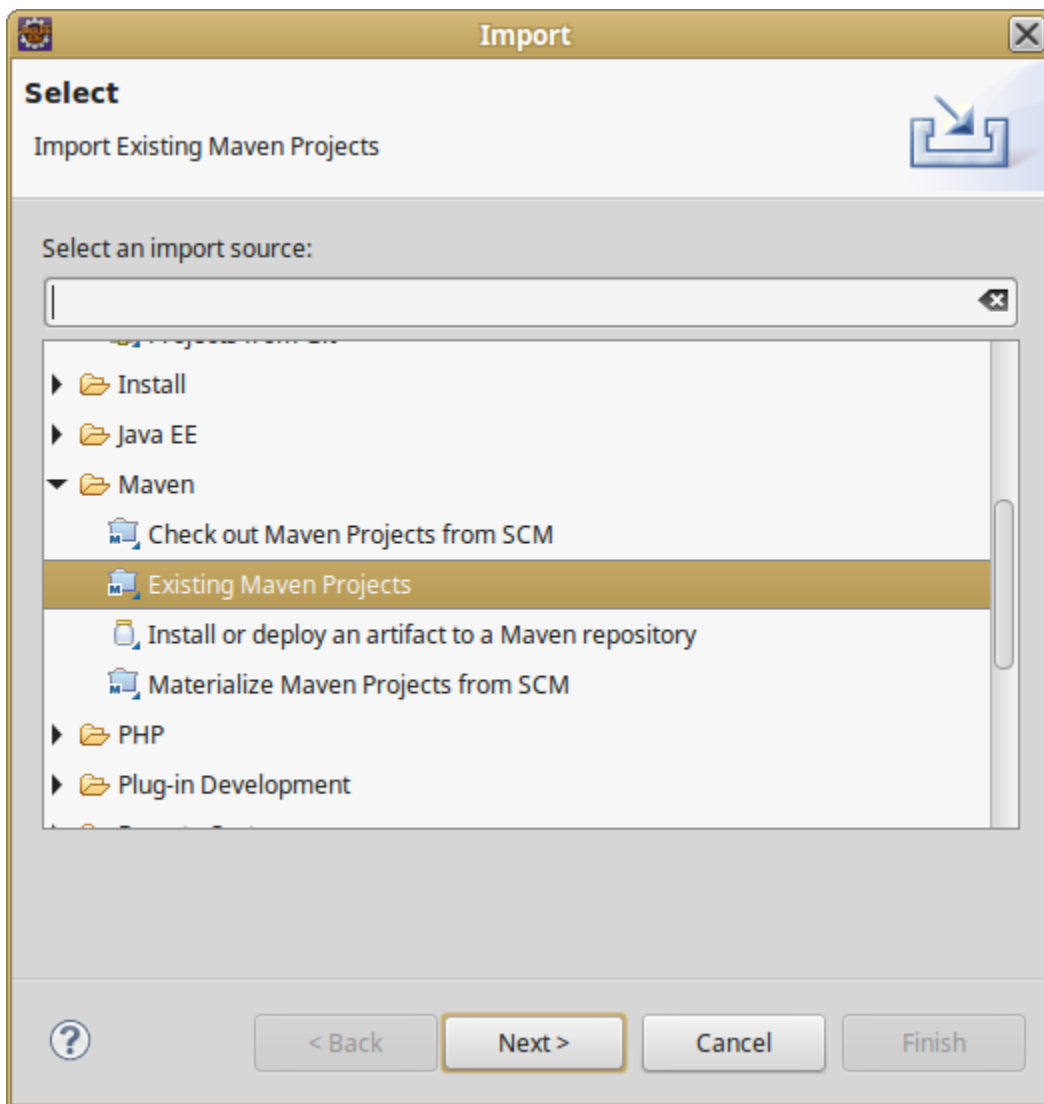
Password:

Store in Secure Store ☒

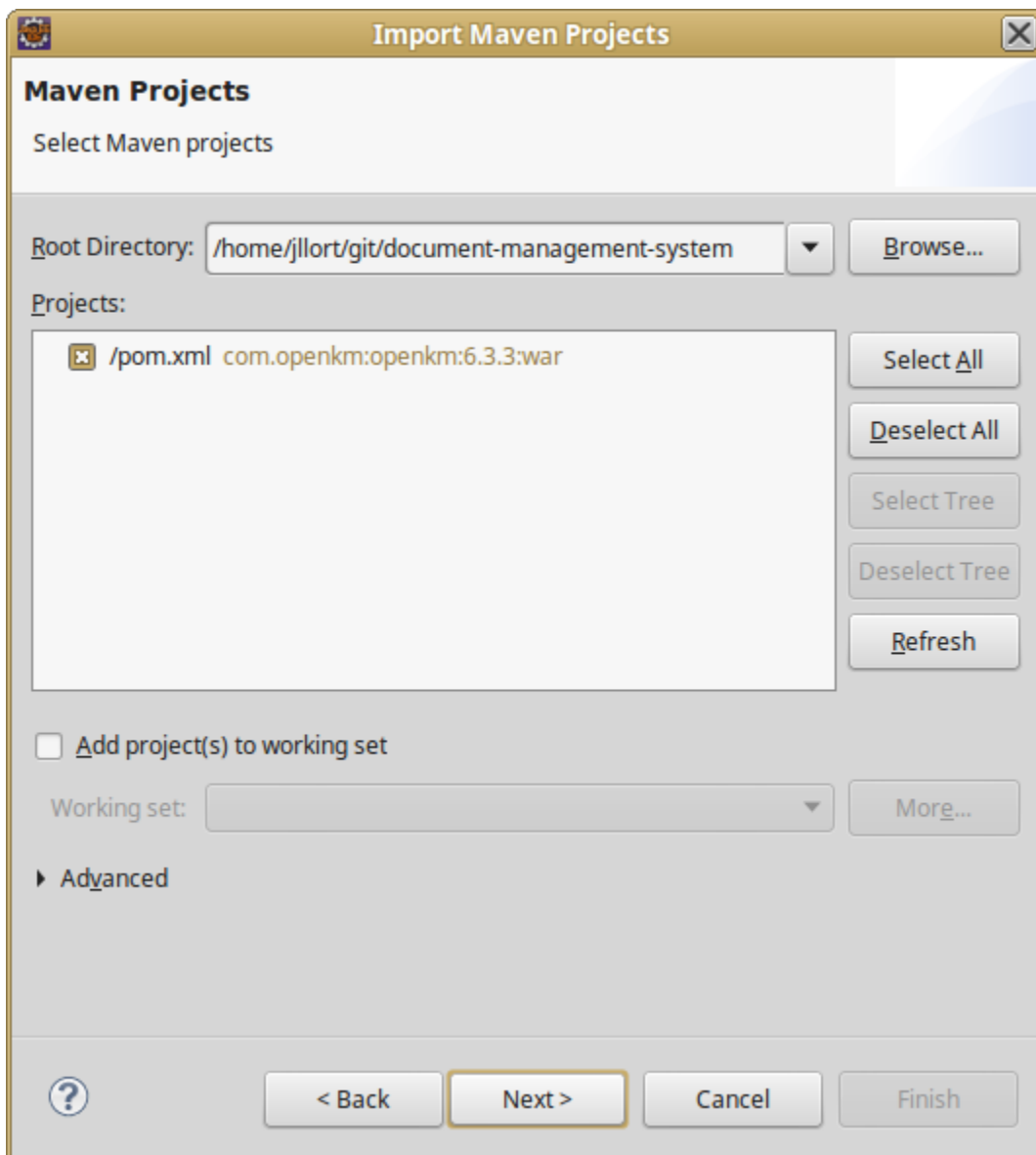
Click on **Next** button.



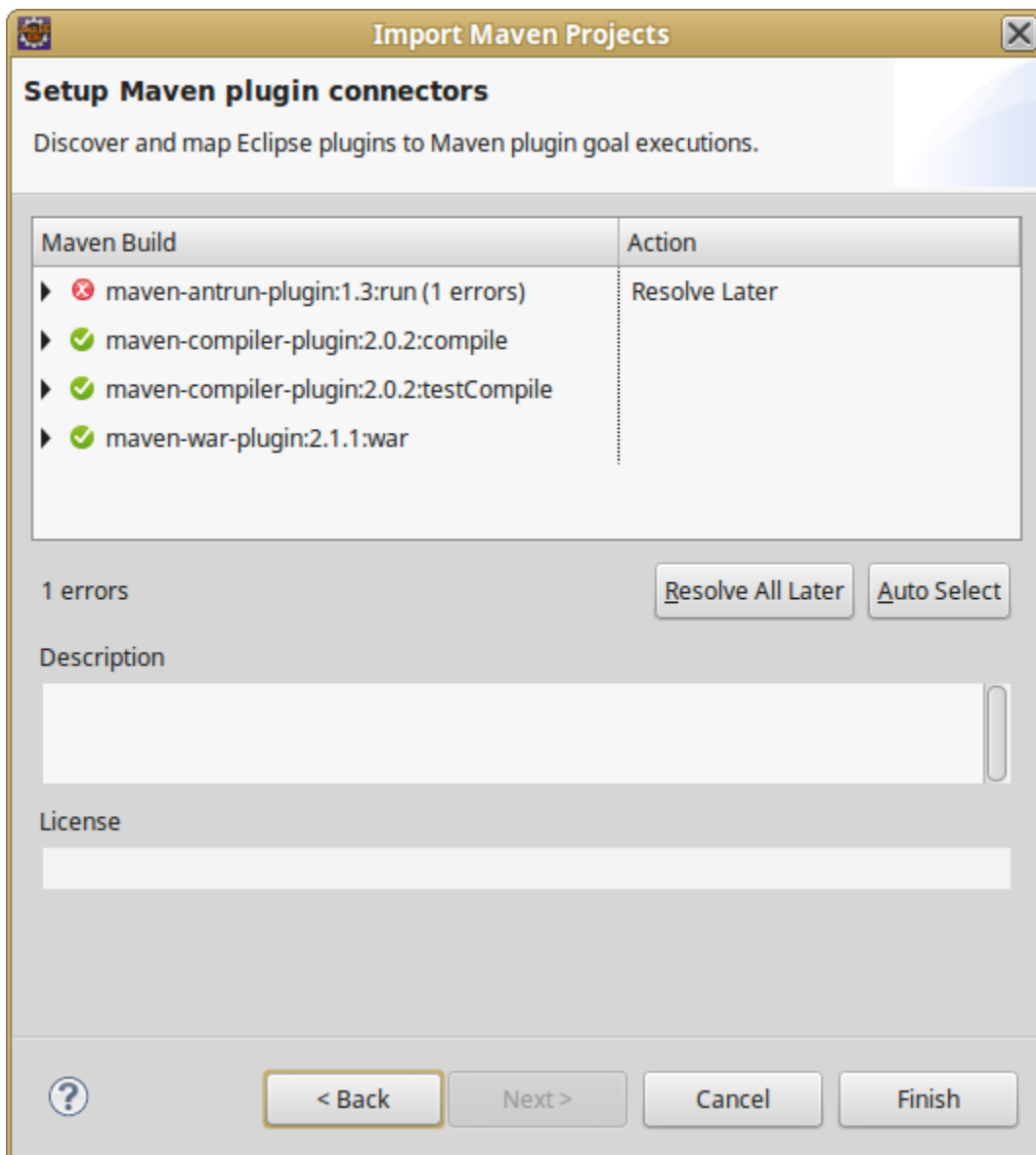
Finally import the project into eclipse. Go to **File > Import > Existing Maven Projects**.



*Choose the **project document-management-system** folder from your local git repository and click on **Next** button.*



Finally click on **Finish** button.



Checkout source code from Subversion

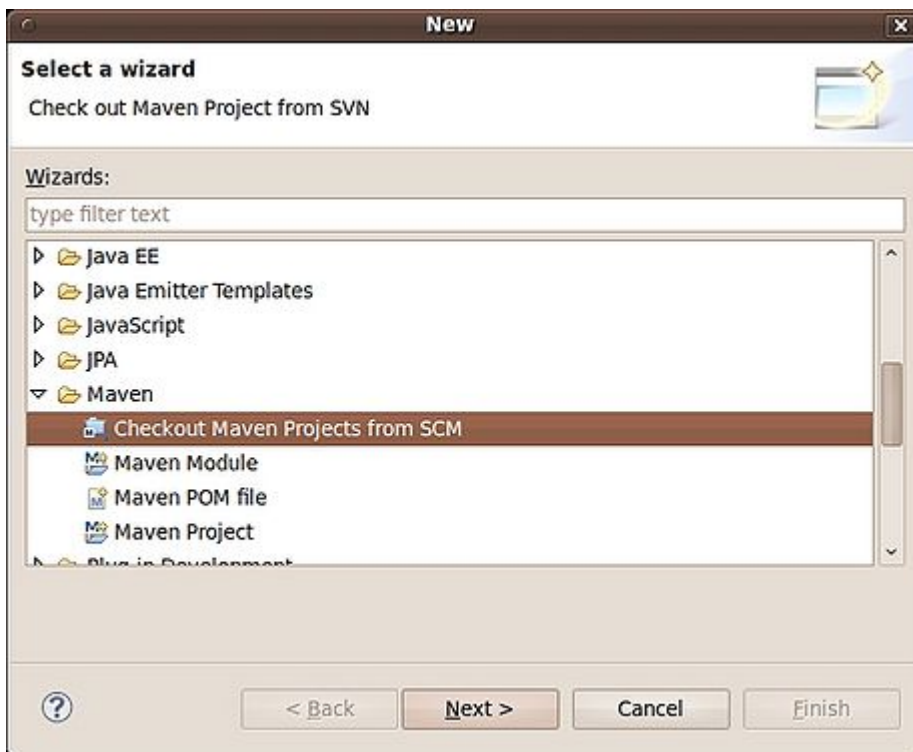


From version 6.3.3 the source code is available at GitHub, take a look at [Checkout source code from Git](#).

Go to **File / New / Other** and select **Maven / Checkout Maven Projects from SCM**:

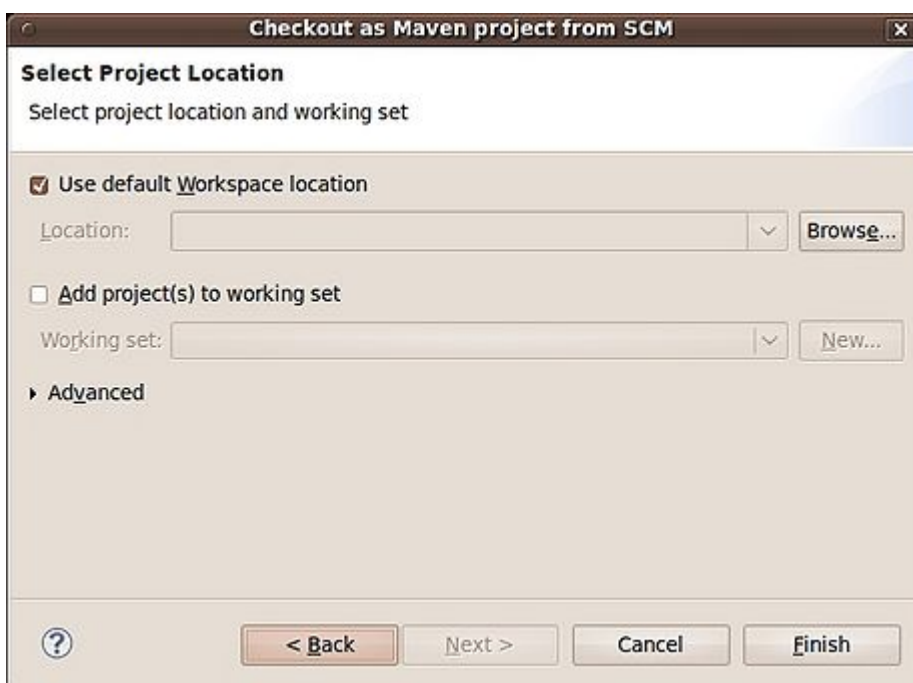
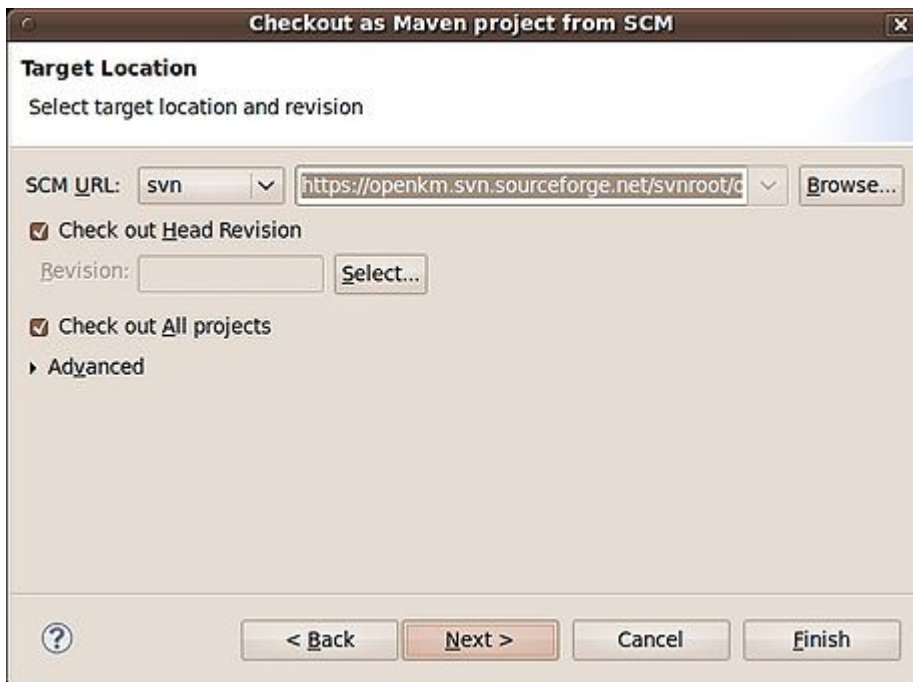


Hint: If your SCM dropdown list is empty, install the Maven Subclipse extension from (<http://m2eclipse.sonatype.org/sites/m2e-extras/>).

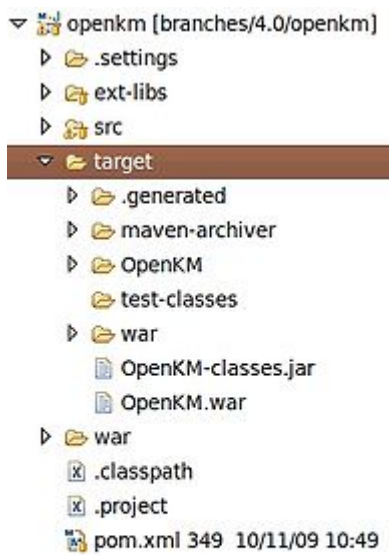


Version	Url	Command line
6.3.x	http://svn.code.sf.net/p/openkm/code/branches/6.3/	\$ svn co http://svn.code.sf.net/p/openkm/code/branches/6.3/
6.2.x	http://svn.code.sf.net/p/openkm/code/branches/6.2/openkm	\$ svn co http://svn.code.sf.net/p/openkm/code/branches/6.2/
5.1.x	http://svn.code.sf.net/p/openkm/code/branches/5.1/openkm	\$ svn co http://svn.code.sf.net/p/openkm/code/branches/5.1/
4.1.x	http://svn.code.sf.net/p/openkm/code/branches/4.1/openkm	\$ svn co

		<code>http://svn.code.sf.net/p/openkm/code/branches/4.0</code>
4.0.x	http://svn.code.sf.net/p/openkm/code/branches/4.0/openkm	<pre>\$ svn co http://svn.code.sf.net/p/openkm/code/branches/4.0</pre>



Be patient, first time you downloading OpenKM from svn, you're downloading all libraries to your maven repository. It could take several minutes depending your Internet connection.



OpenKM will be downloaded, and automatically compiled.



If you want native support for Subversion, you have to install the `libsvn-java` package in Linux. The edit the `eclipse.ini` configuration file and add the following line:

```
-Djava.library.path=/usr/lib/jni
```

Eclipse tips

Debug tomcat

You can debug your OpenKM installation using the Tomcat logging facility. This is an useful thing when you have problems with your configuration. Default OpenKM installation tries to log important events like errors and warnings. Is possible to change this configuration editing the file `$TOMCAT_HOME/conf/logging.properties`.

Default Tomcat log configuration can generate a lot of messages. These files are stored at `$TOMCAT_HOME/logs`.

Debug remote tomcat server



When you have problems on the production environment this configuration is specially useful.

Add the following options when the JVM is started:

```
-Xdebug -Xrunjdwp:transport=dt_socket,address=8000,server=y,suspend=n
```

Or if you are using shell scripts to start Tomcat, start it with the following command (**recommended**):

```
$ $TOMCAT_HOME/bin/catalina.sh jpda start
```

It will start Tomcat so that a remote debugger can be connected to port 8000.



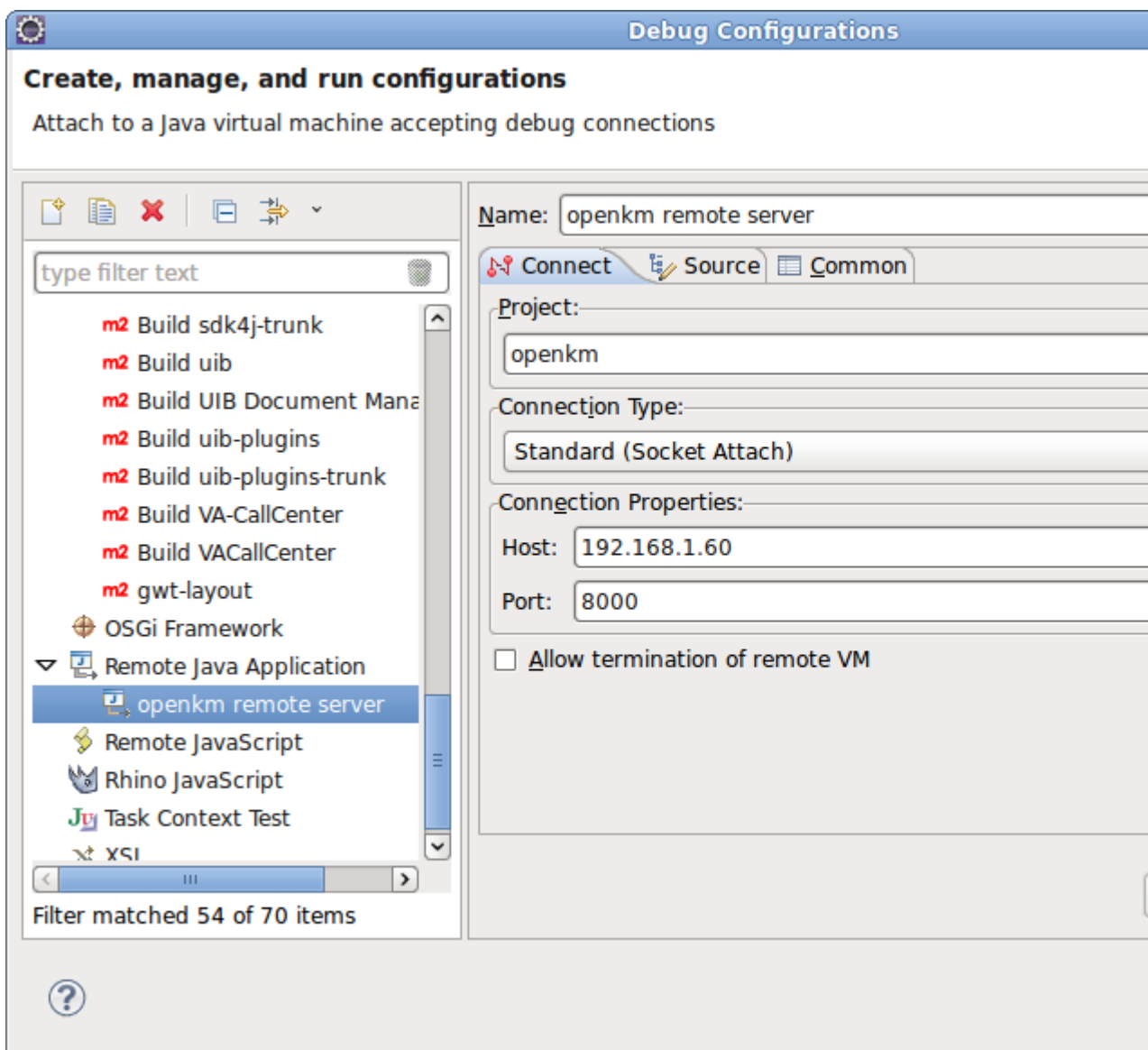
More information at:

- [Tomcat developing: How do I configure Tomcat to support remote debugging?](#)
- [Como hacer debug remoto en Tomcat desde Eclipse.](#)

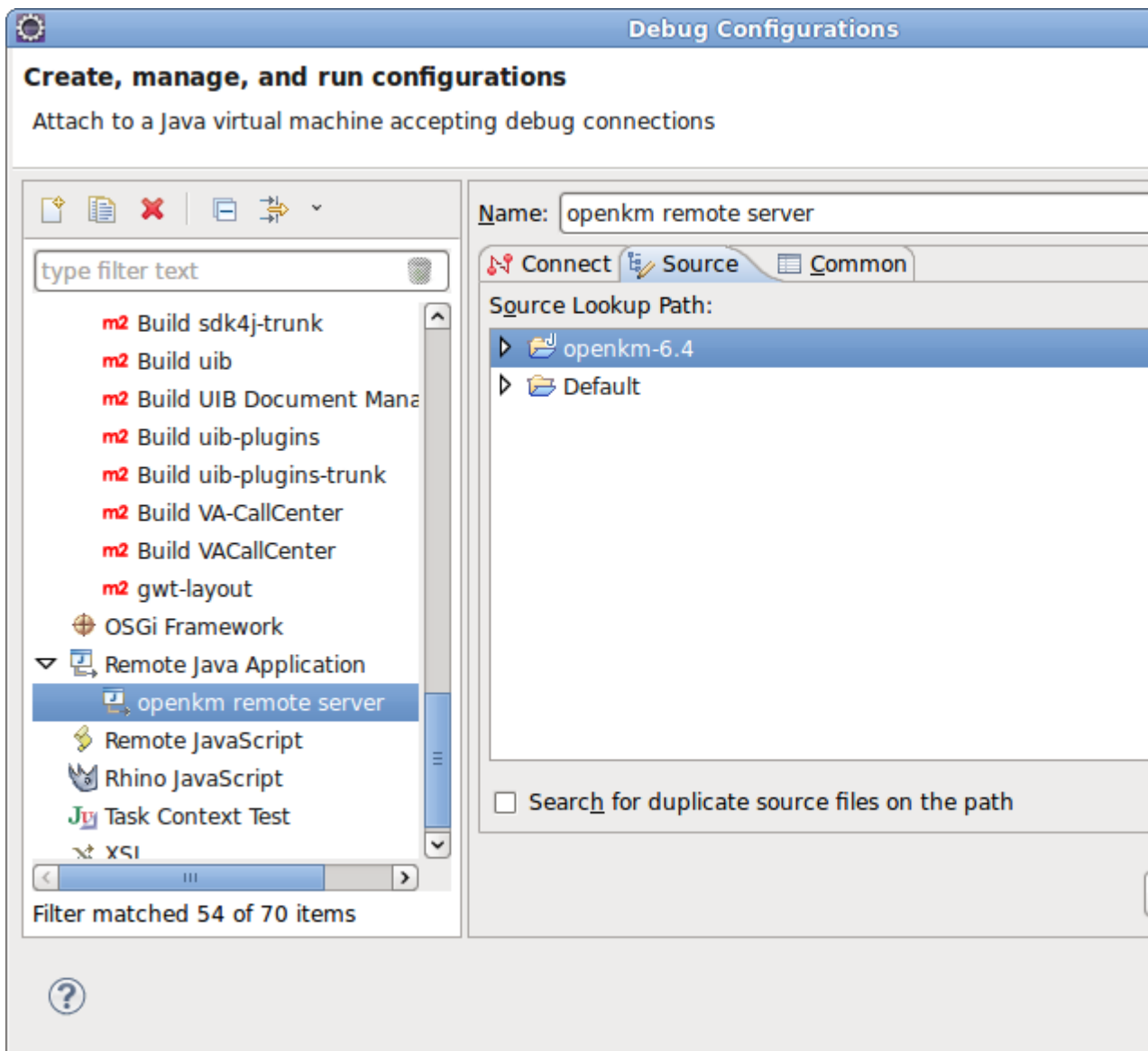
Eclipse configuration

Open eclipse "**Debug configurations**", then choose "**Remote Java Application**", right click on mouse and choose "**new**".

Set the "**host**" and "**port**" of the remote server:



Include in source tab your OpenKM java project:



You can debug normally, setting breakpoints, etc.

Debug GWT frontend user interface with GWT Super Dev Mode



Super Dev Mode is available from version GWT 2.5.0.RC1, more information at [GWT 2.5.0.RC1 Release notes](#).

You can find additional information of them at :

- [GWT Super Dev Mode official site](#).

Enable source for debugging

Edit `pom.xml` file and uncomment this section:

```
<!-- Only for development -->
<dependency>
  <groupId>com.google.gwt</groupId>
  <artifactId>gwt-dev</artifactId>
  <version>${gwt.version}</version>
  <scope>provided</scope>
</dependency>
```

Uncomment the xsiframe section into the Main.gwt.xml file:

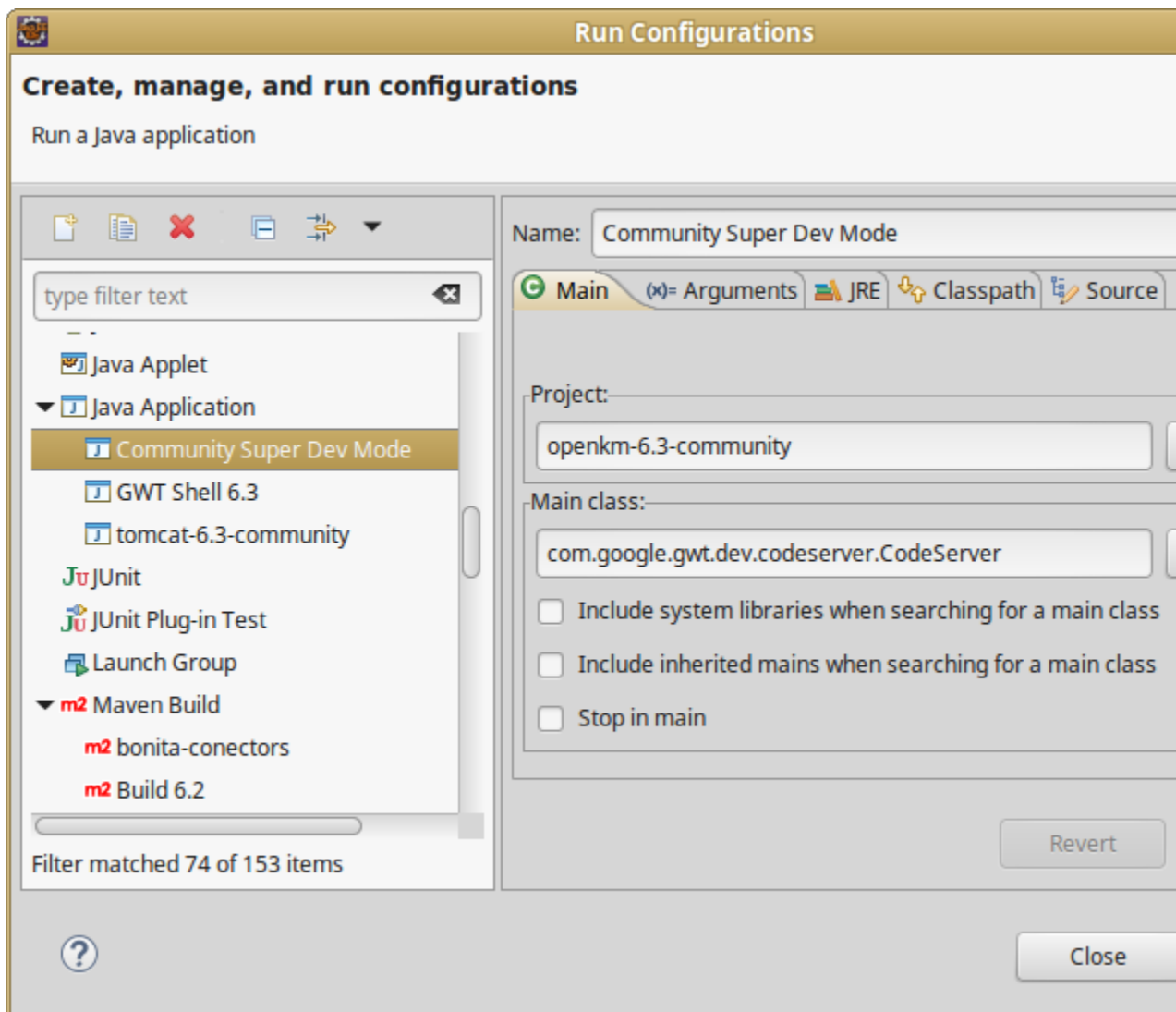
```
<!--
  Use GWT 2.7.0 to debug in Firefox -->
  <add-linker name="xsiframe"/>
  <set-configuration-property name='xsiframe.failIfScriptTag' value='FALSE'/>
  <set-configuration-property name="devModeUrlWhitelistRegexp" value="http://(localhost|127.0.0.1)"/>
```

Eclipse configuration

Open eclipse "**Debug configurations**", then choose "**Java Application**", right click on mouse and choose "**new**".

Set the Main class field value:

```
com.google.gwt.dev.DevMode
```

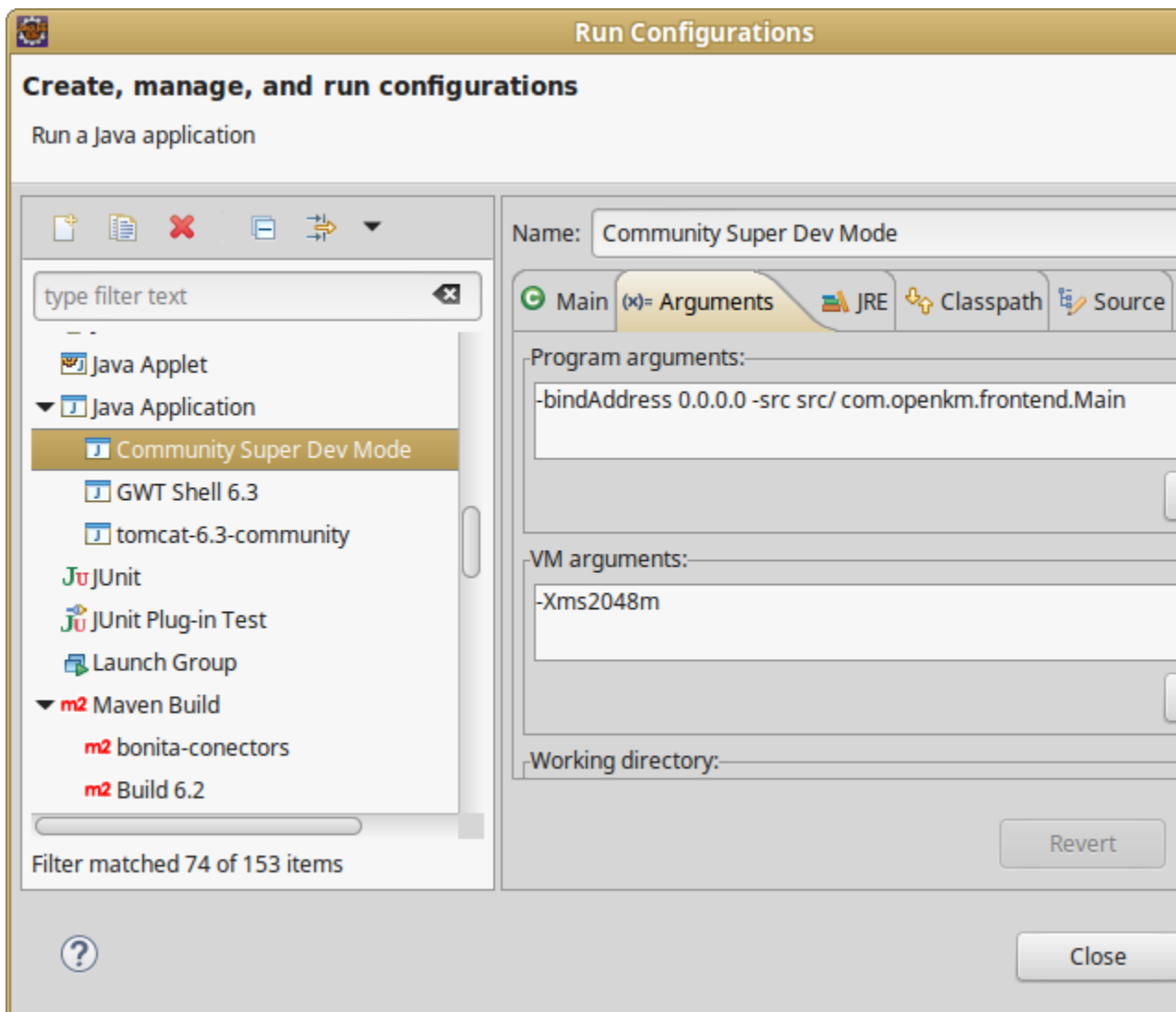


Select "**Arguments**" tab and set the "**program arguments**" field value:

```
-bindAddress 0.0.0.0 -src src/ com.openkm.frontend.Main
```

And the "**VM arguments**" field value:

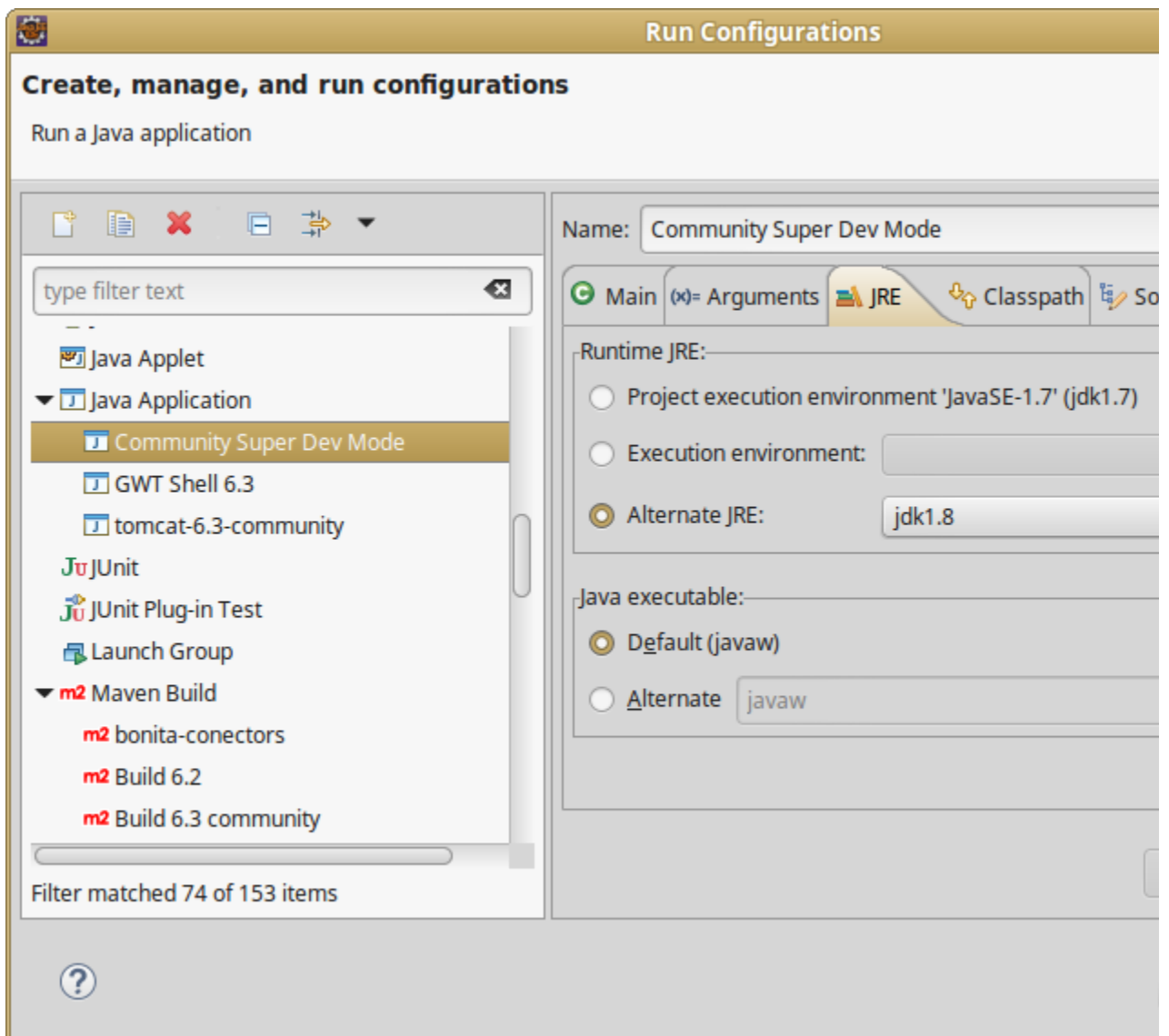
```
-Xms2048m
```



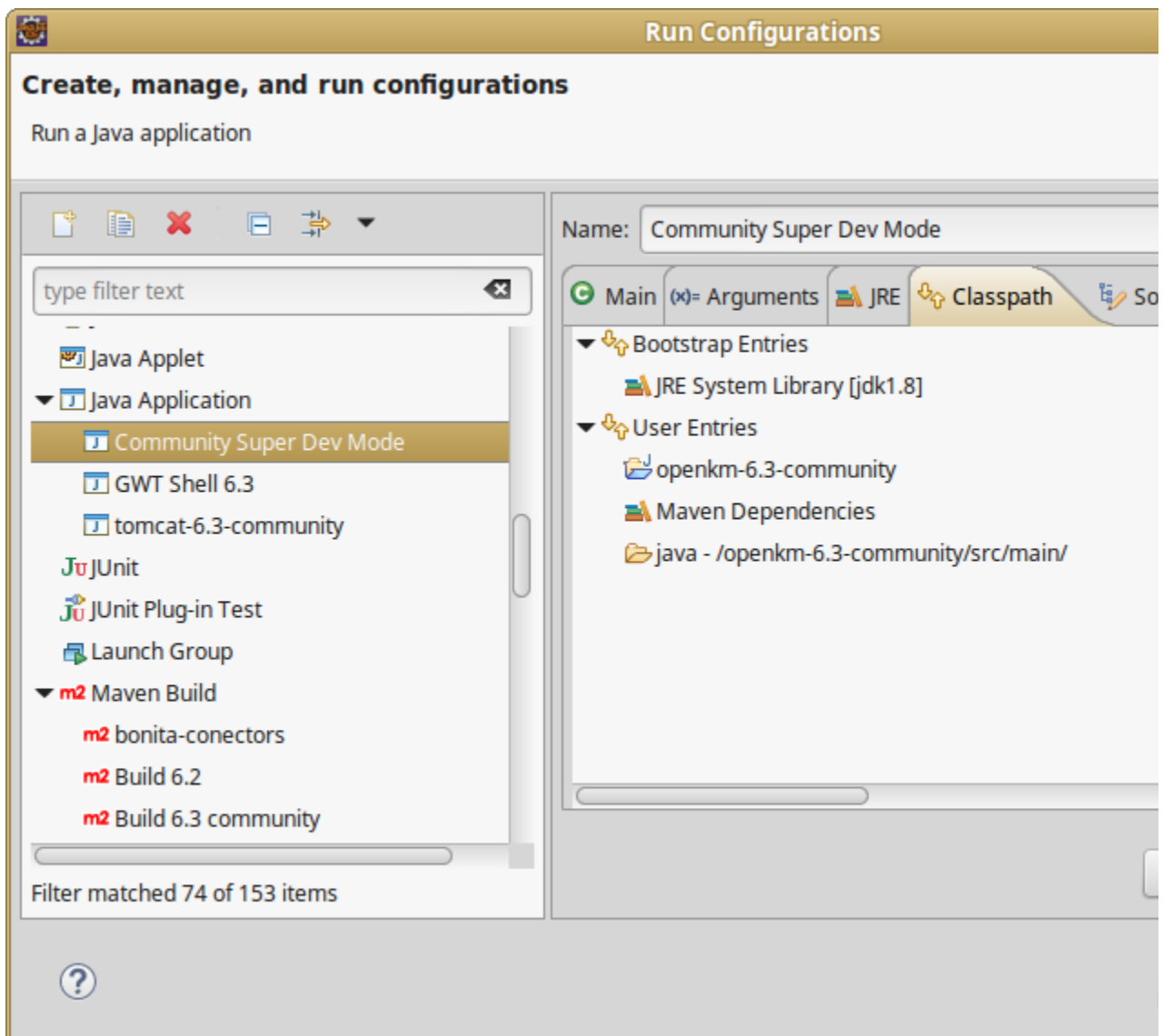
Select "**JRE**" tab and set the "**OpenJDK version**":



You should choose OpenJDK 1.8.x



Select the "**Classpath**" tab and add folder **/openkm/src/main** (use "Advanced" button and "add folder" option for it) :



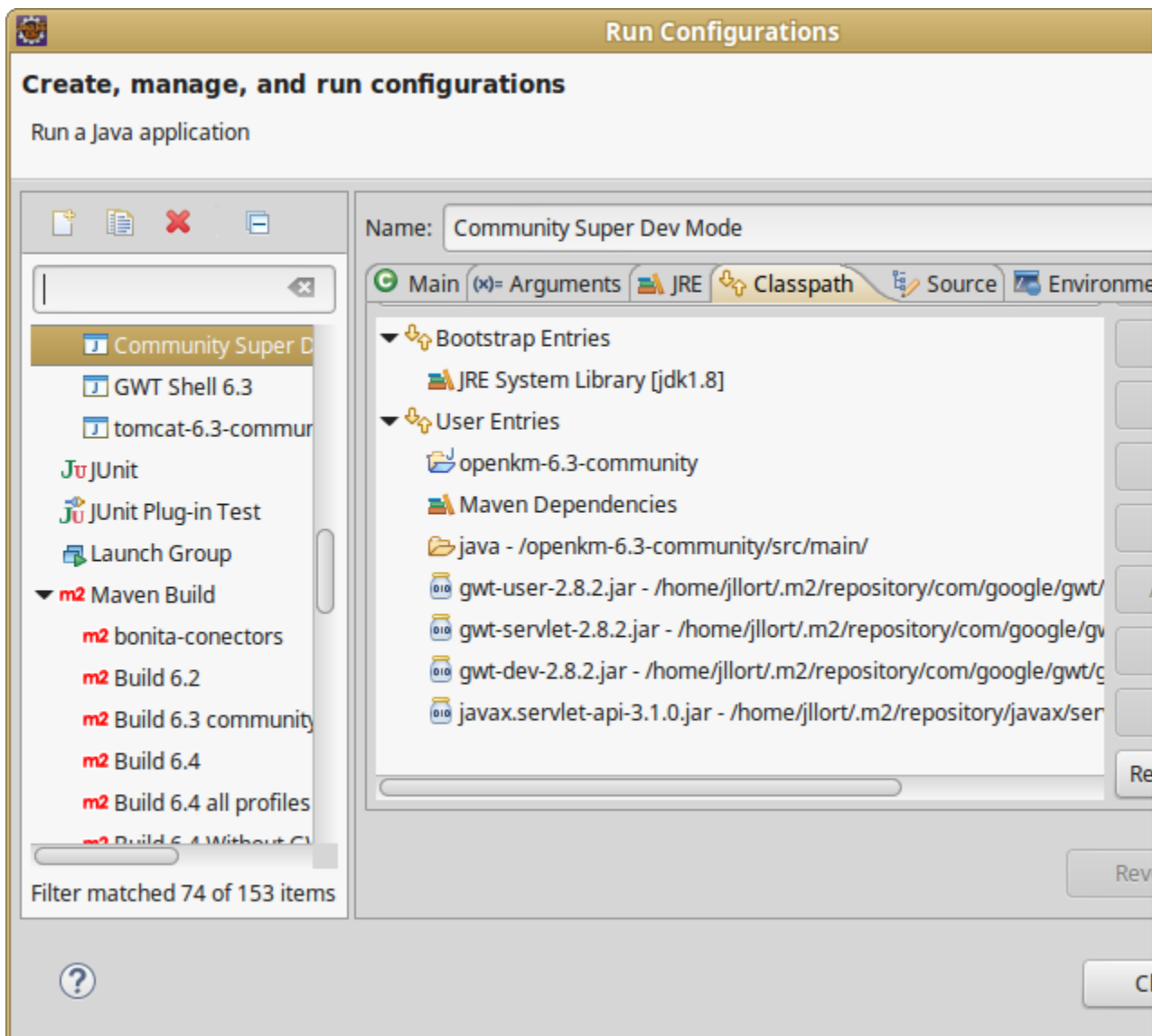
Add GWT jar dependencies with Add External JARs button:



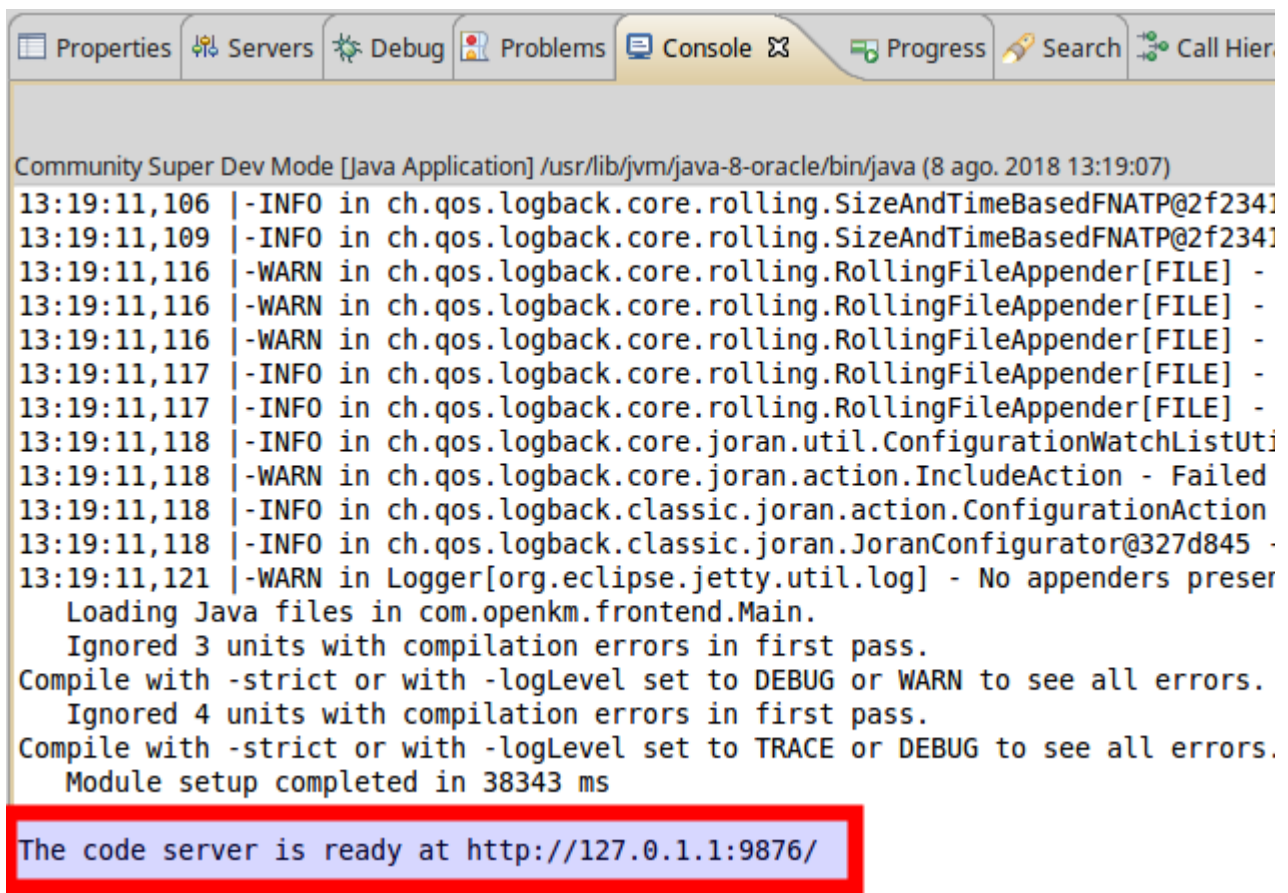
The version of JARs files below might change.

The javax.servlet-api-3.1.0 solve a java.lang.NoClassDefFoundError: javax/servlet/http/HttpSessionIdListener error. That happens from version 2.8.x and upper.

- gwt-user-2.8.2.jar
- gwt-dev-2.8.2.jar
- gwt-user-2.8.2.jar
- javax.servlet-api-3.1.0



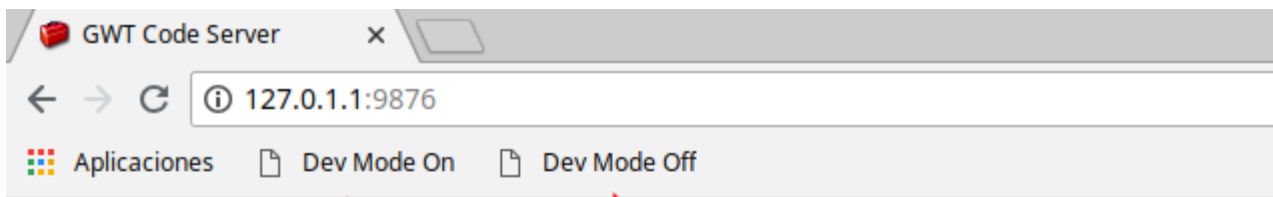
No you can **run the Super Dev Mode** at the end of the log is shown the url where the service is binding:



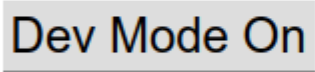
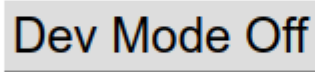
```
Community Super Dev Mode [Java Application] /usr/lib/jvm/java-8-oracle/bin/java (8 ago. 2018 13:19:07)
13:19:11,106 |-INFO in ch.qos.logback.core.rolling.SizeAndTimeBasedFNATP@2f2341
13:19:11,109 |-INFO in ch.qos.logback.core.rolling.SizeAndTimeBasedFNATP@2f2341
13:19:11,116 |-WARN in ch.qos.logback.core.rolling.RollingFileAppender[FILE] -
13:19:11,116 |-WARN in ch.qos.logback.core.rolling.RollingFileAppender[FILE] -
13:19:11,116 |-WARN in ch.qos.logback.core.rolling.RollingFileAppender[FILE] -
13:19:11,117 |-INFO in ch.qos.logback.core.rolling.RollingFileAppender[FILE] -
13:19:11,117 |-INFO in ch.qos.logback.core.rolling.RollingFileAppender[FILE] -
13:19:11,118 |-INFO in ch.qos.logback.core.joran.util.ConfigurationWatchListUtil
13:19:11,118 |-WARN in ch.qos.logback.core.joran.action.IncludeAction - Failed
13:19:11,118 |-INFO in ch.qos.logback.classic.joran.action.ConfigurationAction
13:19:11,118 |-INFO in ch.qos.logback.classic.joran.JoranConfigurator@327d845 -
13:19:11,121 |-WARN in Logger[org.eclipse.jetty.util.log] - No appenders preser
Loading Java files in com.openkm.frontend.Main.
Ignored 3 units with compilation errors in first pass.
Compile with -strict or with -logLevel set to DEBUG or WARN to see all errors.
Ignored 4 units with compilation errors in first pass.
Compile with -strict or with -logLevel set to TRACE or DEBUG to see all errors.
Module setup completed in 38343 ms

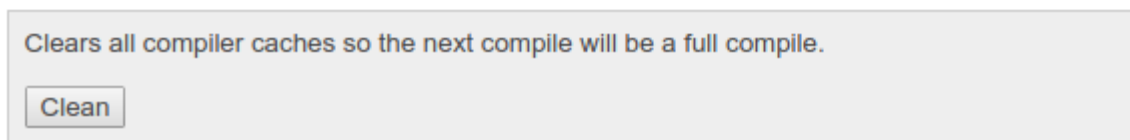
The code server is ready at http://127.0.1.1:9876/
```

Open the **code server** url in your **chrome** browser browser and drag the **bookmarklets** in your **bookmark bar** :

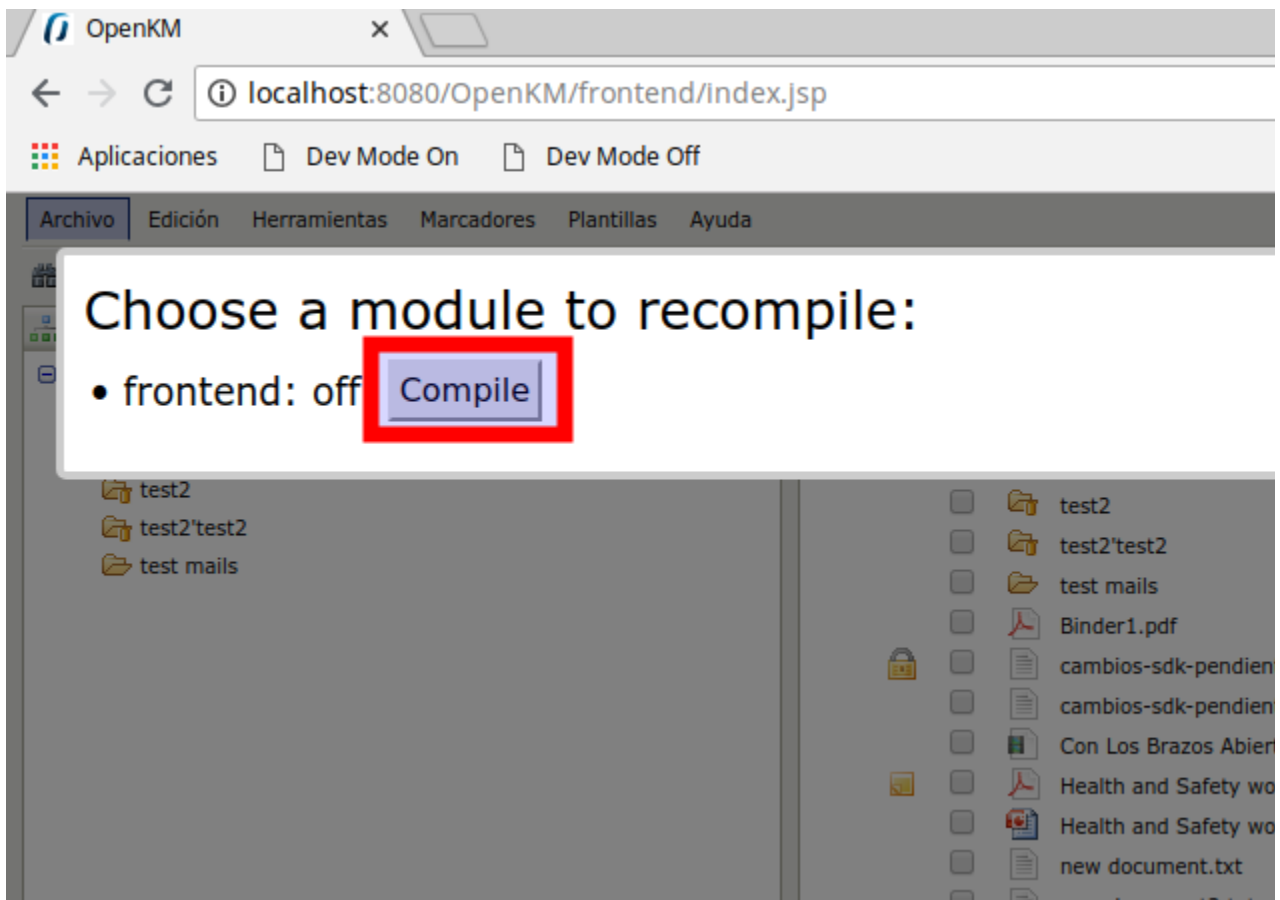


GWT Code Server

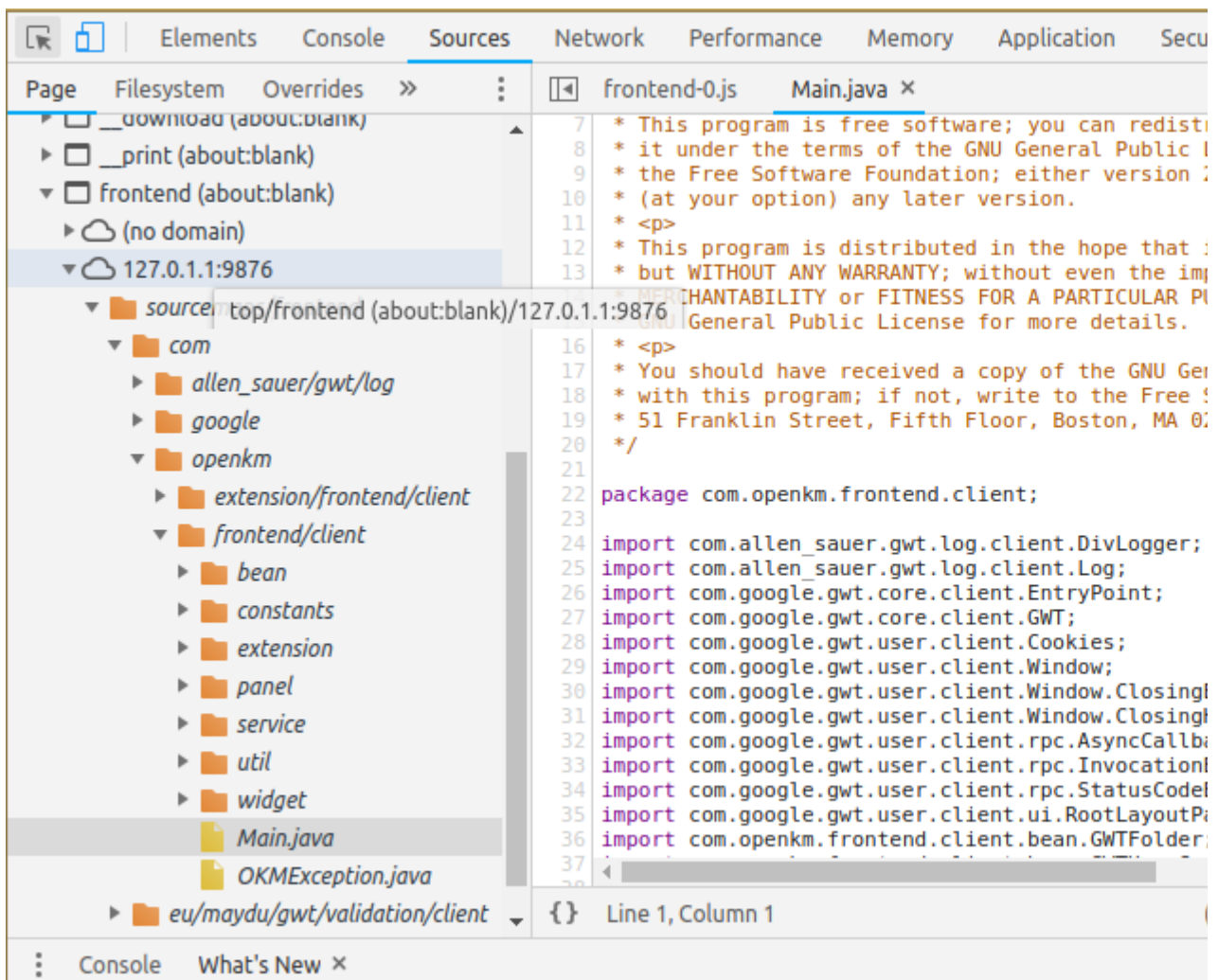
1. Drag these two bookmarklets to your browser's bookmark bar:
 
2. Visit a web page that uses one of these modules:
 - frontend
3. Click "Dev Mode On" to start development.



Open your **OpenKM frontend** url in **Chrome browser** and then **click in your bookmark bar the option "Dev Mode On"** you have dragged in previous step. **It will be shown a popup option with Compile button. Click in "Compile" button.**



Finally **open** the **Chrome developer tools** and from the **sources tab** you have OpenKM java code available, where to set breakpoints, watch variables, etc:



Debug GWT frontend user interface with GWT Shell



GWT Shell is deprecated as it has been removed from GWT version 2.5.1, more information at [GWT 2.5.1 Release notes](#).



You can debug GWT frontend with GWT Shell (what have extension support from firefox browsers 2.4 and older) or GWT Super Dev Mode (we suggest use chrome in this scenario).

In the documentation section below, we have tried to explain how to get working old GWT Shell with eclipse. The sample below show how to debug GWT from localhost, but also is possible debug remotely.

You can find additional information of them at :

- [GWT Shell official site](#).

Enable source for debugging

Edit `pom.xml` file and uncomment this section:

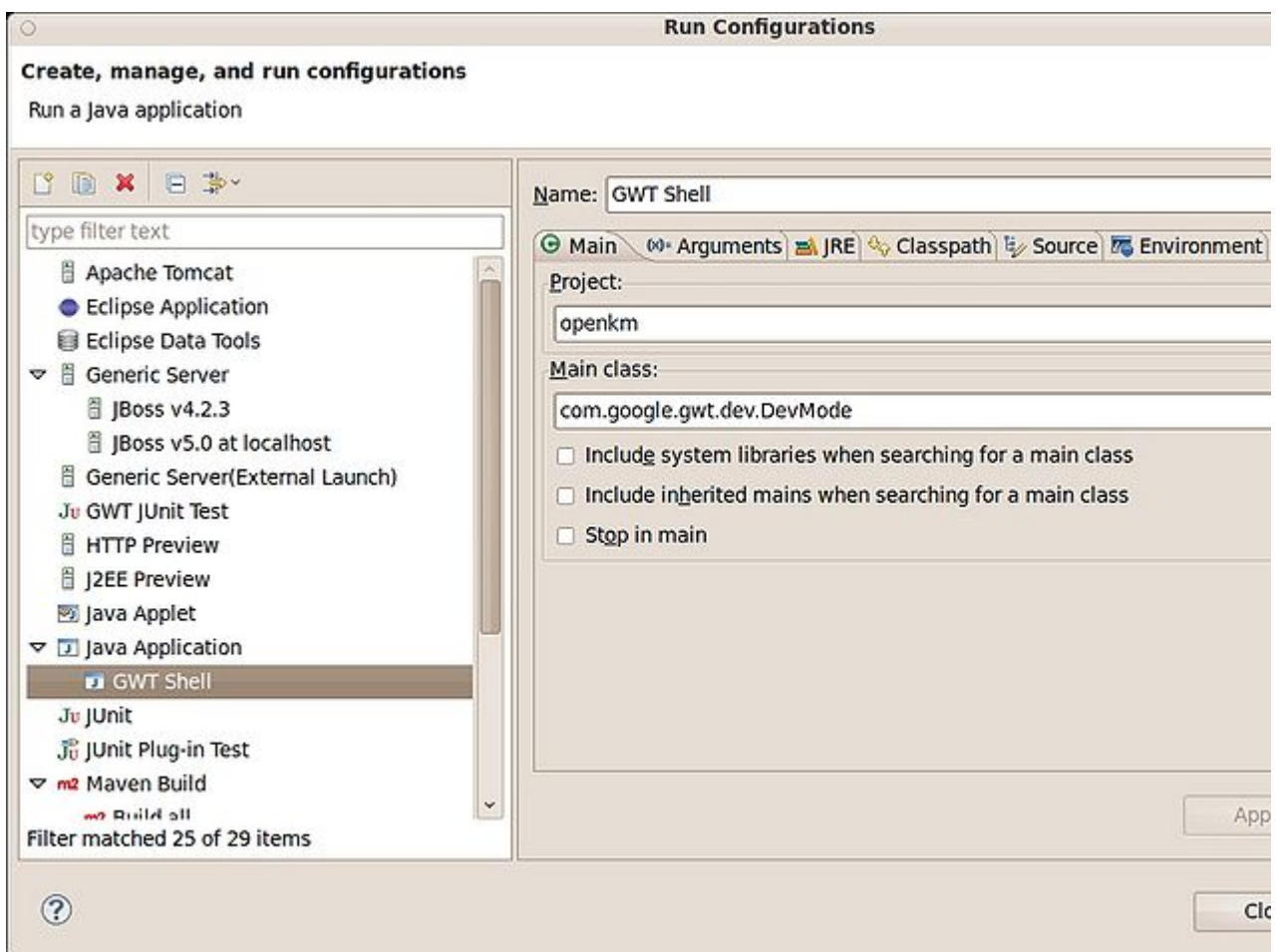
```
<!-- Only for development -->
<dependency>
  <groupId>com.google.gwt</groupId>
  <artifactId>gwt-dev</artifactId>
  <version>${gwt.version}</version>
  <scope>provided</scope>
</dependency>
```

Eclipse configuration

Open eclipse "**Debug configurations**", then choose "**Java Application**", right click on mouse and choose "**new**".

Set the Main class field value:

```
com.google.gwt.dev.DevMode
```

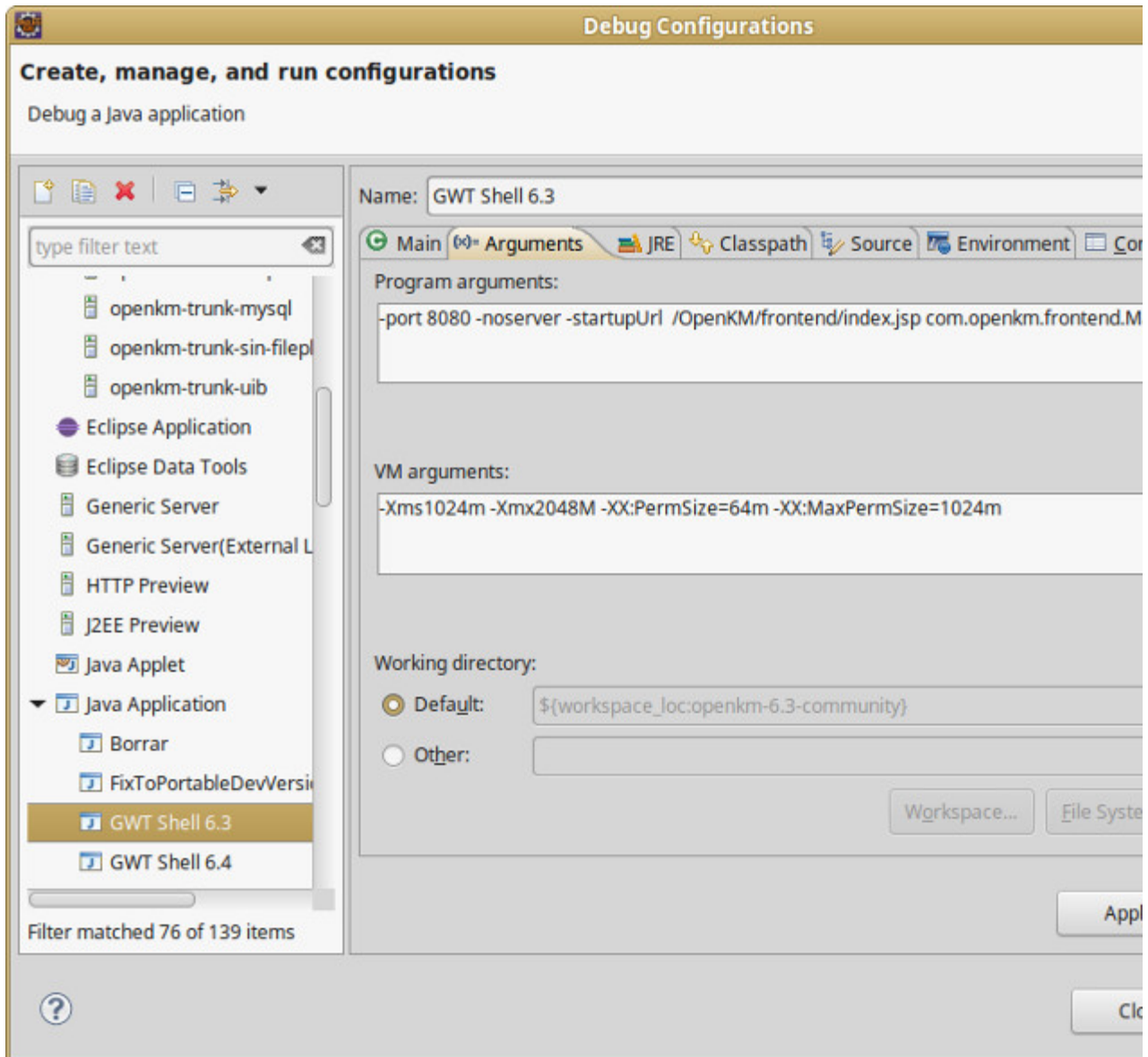


Select "**Arguments**" tab and set the "**program arguments**" field value:

```
-port 8080 -noserver -startupUrl /OpenKM/frontend/index.jsp com.openkm.frontend.Main
```


And the "**VM arguments**" field value:

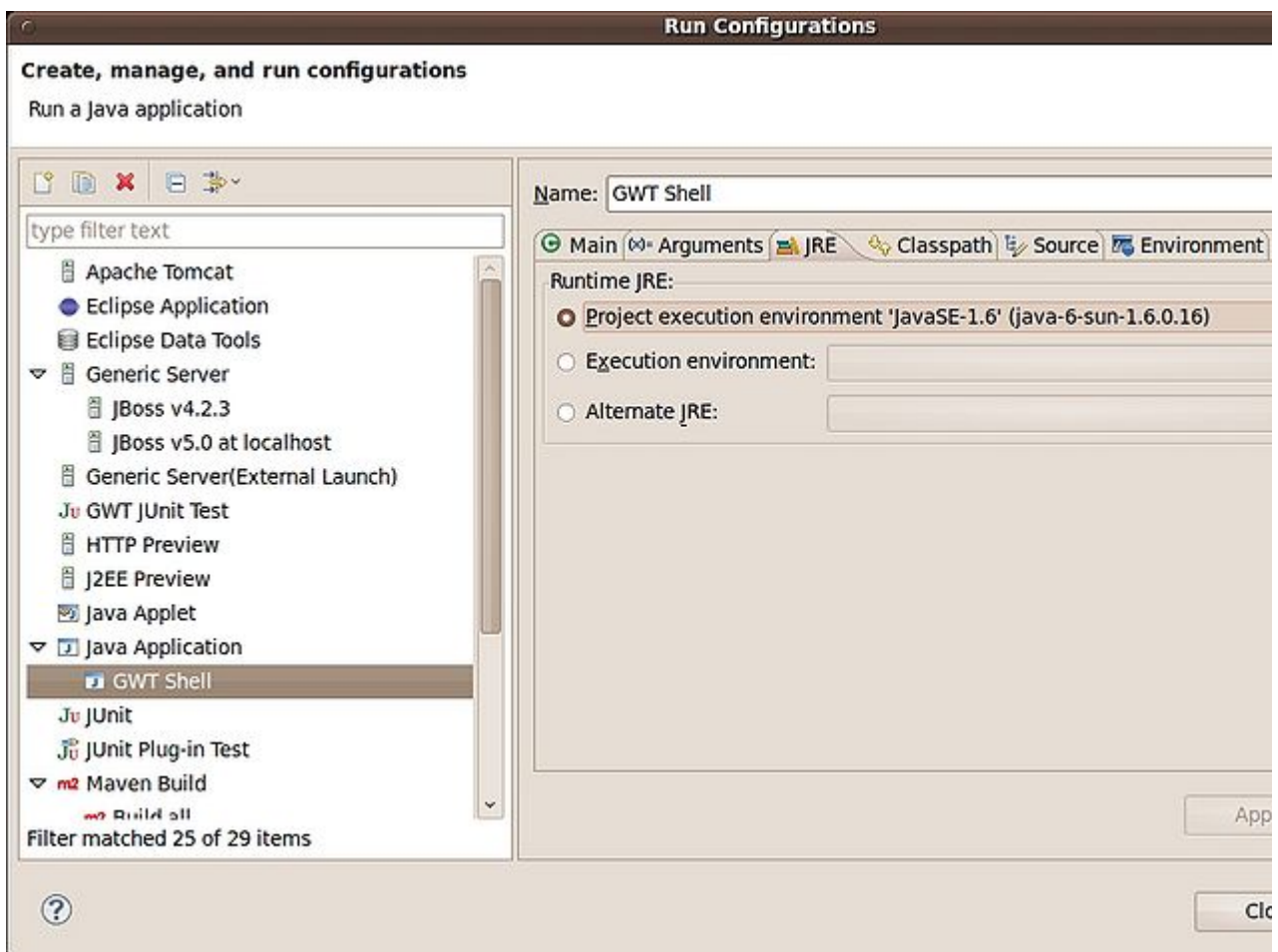
```
-Xms1024m -Xmx2048M -XX:PermSize=64m -XX:MaxPermSize=1024m
```



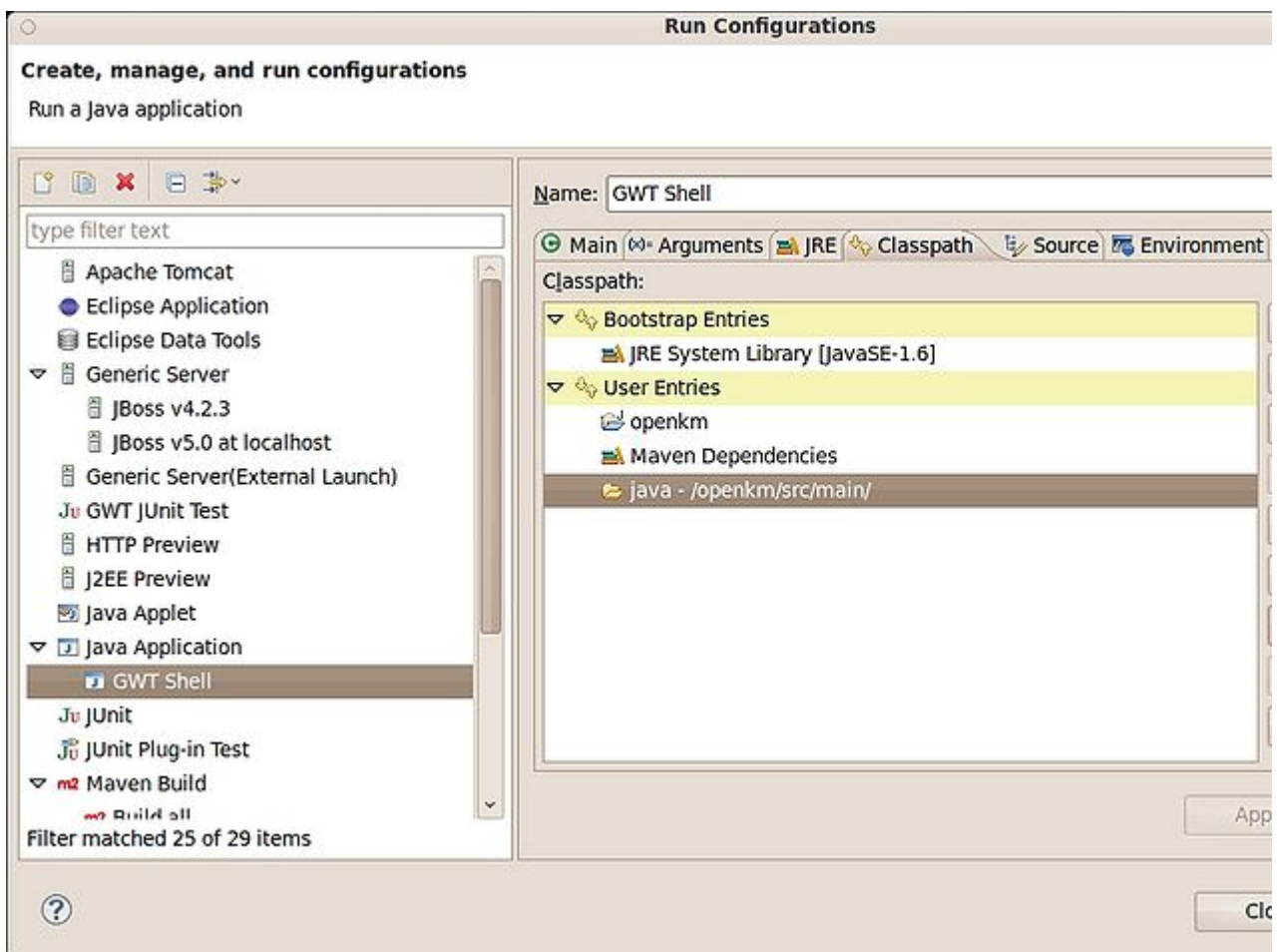
Select "**JRE**" tab and set the "**OpenJDK version**":



You should choose OpenJDK 1.8.x



Select the "**Classpath**" tab and add folder `/openkm/src/main` (use "Advanced" button and "add folder" option for it) :



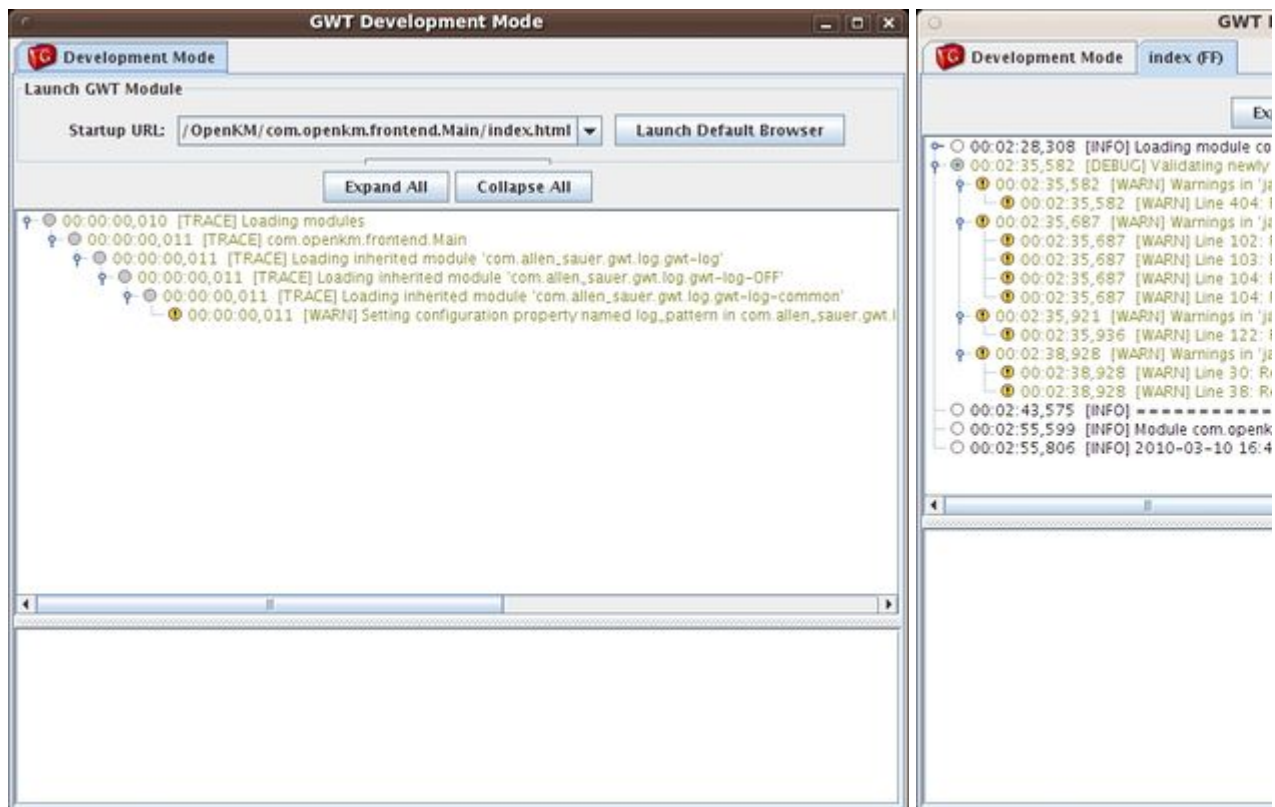
Now you can run the GWT Shell.

To **debug login** put in your browse:

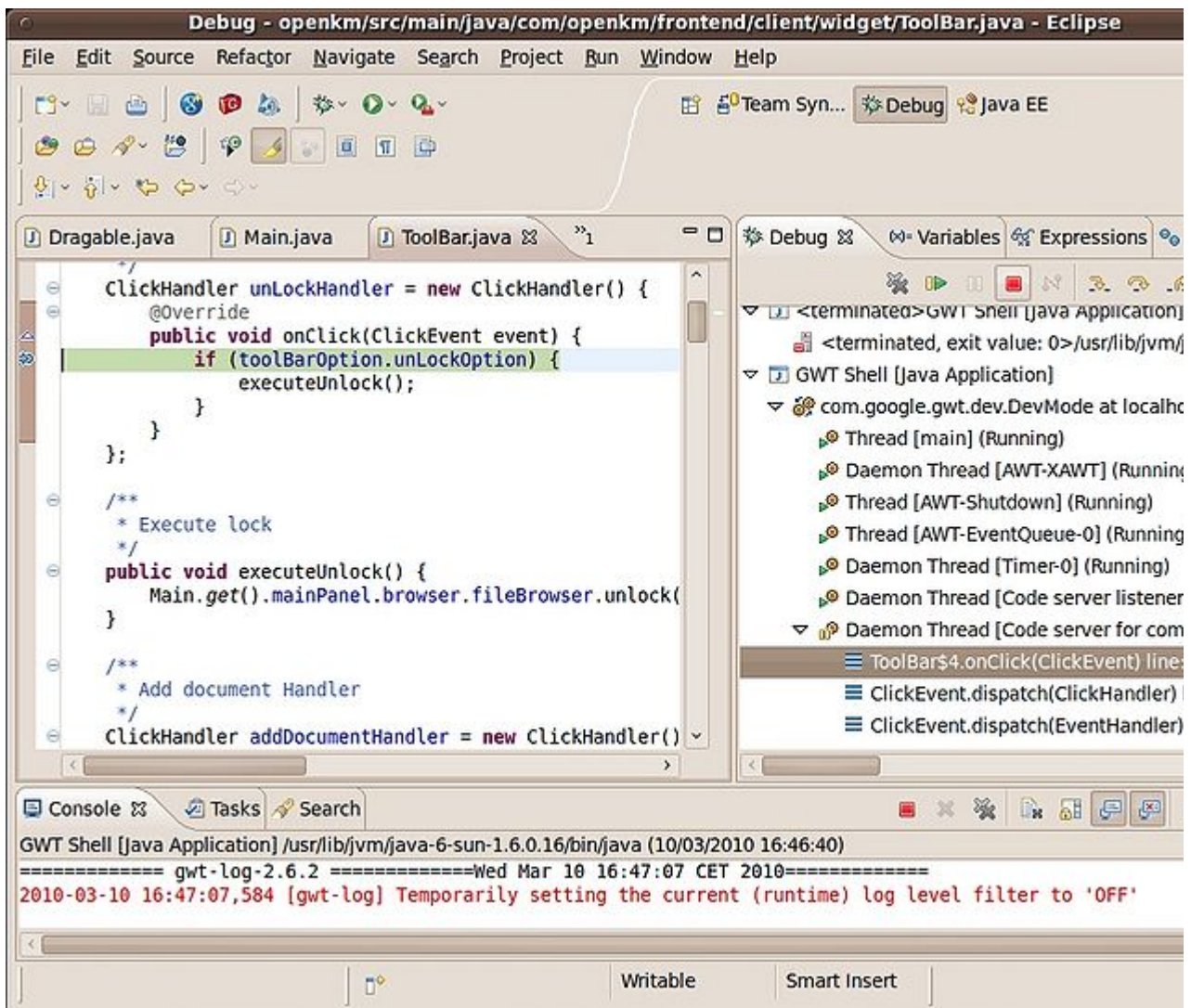


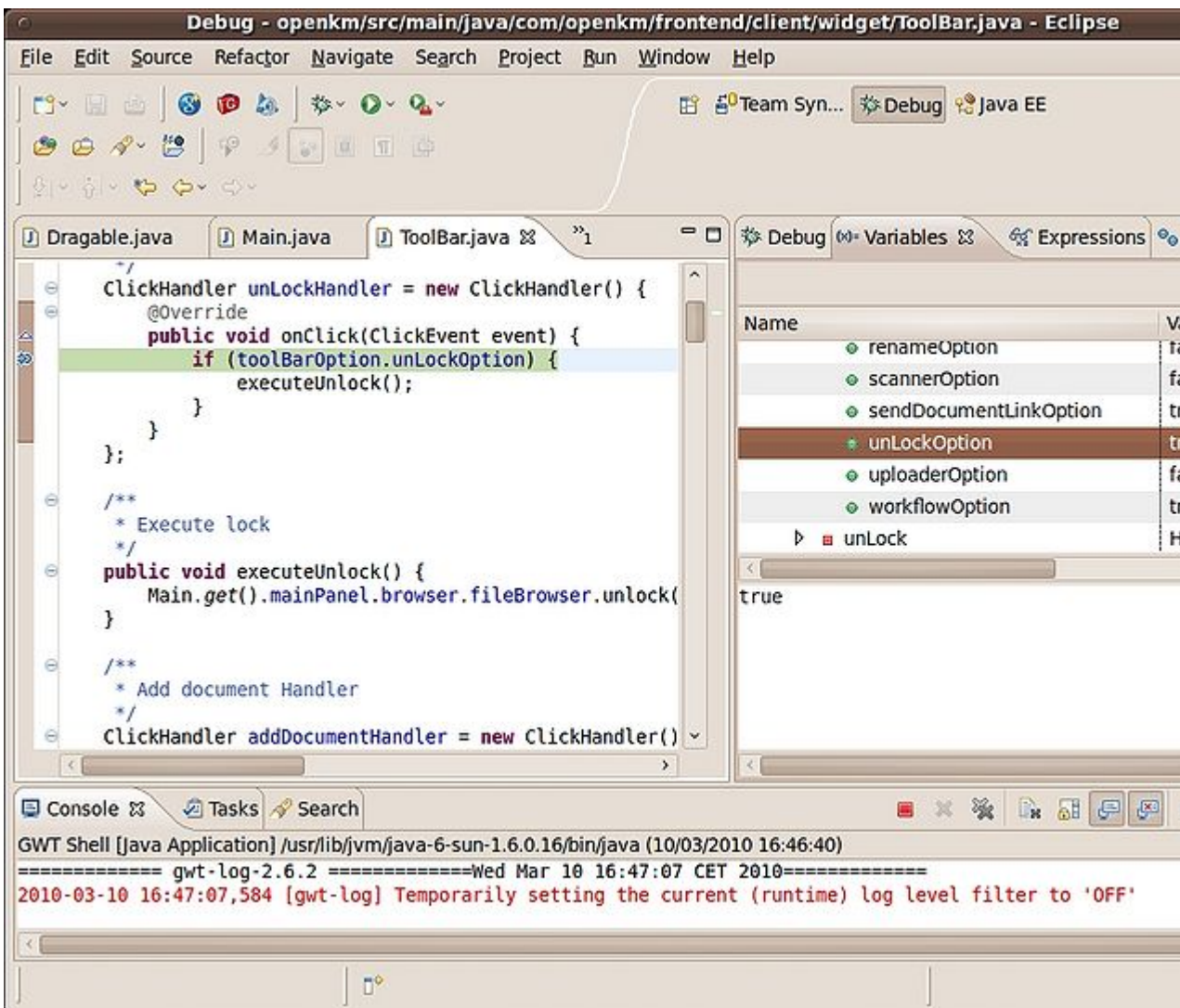
You can remotely debug GWT simply changing localhost for your remote URL. Take in consideration you are executing GWT code from eclipse, not the remote one. Consider your local GWT code version be the same is compiled on the remote server.

```
http://localhost:8080/OpenKM/frontend/index?gwt.codesvr=127.0.1.1:9997
```



You debug into eclipse, see variables values, etc.





Eclipse know troubleshooting

Maven raise error 206 the filename or extension is too long

When you build an application on windows you can get the error **create process error 206 the filename or extension is too long** this is caused to a Windows OS limitation. To solve it:

- Move **.m2** repository to **c:**
- Go to **Eclipse > Windows/Preferences/Maven/User Settings -> Create your own setting.xml** with its content:

```
<settings>
  <localRepository>c:/.m2/repository</localRepository>
</settings>
```

Development tips

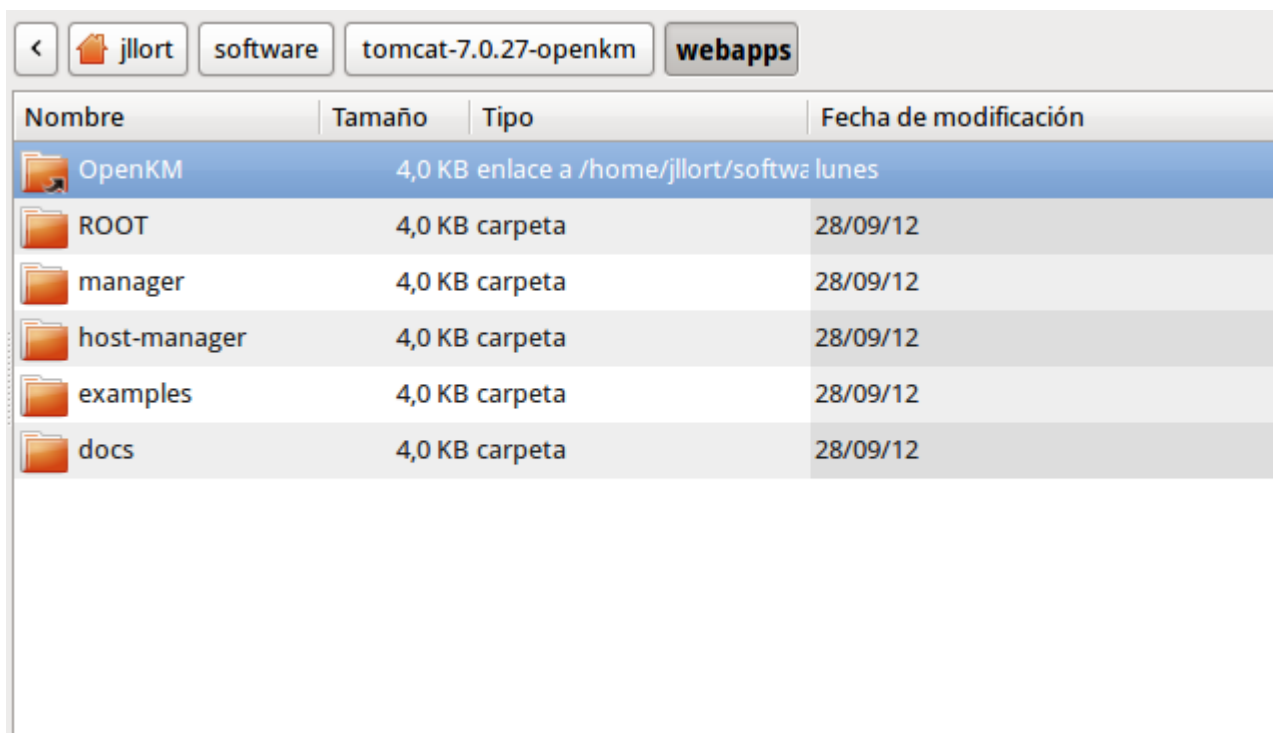
Tomcat application deployment

For faster development we recommend not deploying `OpenKM.war` directly, file and create in `$TOMCAT_HOME/webapps` a folder alias of `target/OpenKM`:



To create symbolic links in Windows use "**junction.exe**" application.

More information at <http://technet.microsoft.com/en-us/sysinternals/bb896768>.



Nombre	Tamaño	Tipo	Fecha de modificación
OpenKM	4,0 KB	enlace a /home/jllort/software	lunes
ROOT	4,0 KB	carpeta	28/09/12
manager	4,0 KB	carpeta	28/09/12
host-manager	4,0 KB	carpeta	28/09/12
examples	4,0 KB	carpeta	28/09/12
docs	4,0 KB	carpeta	28/09/12

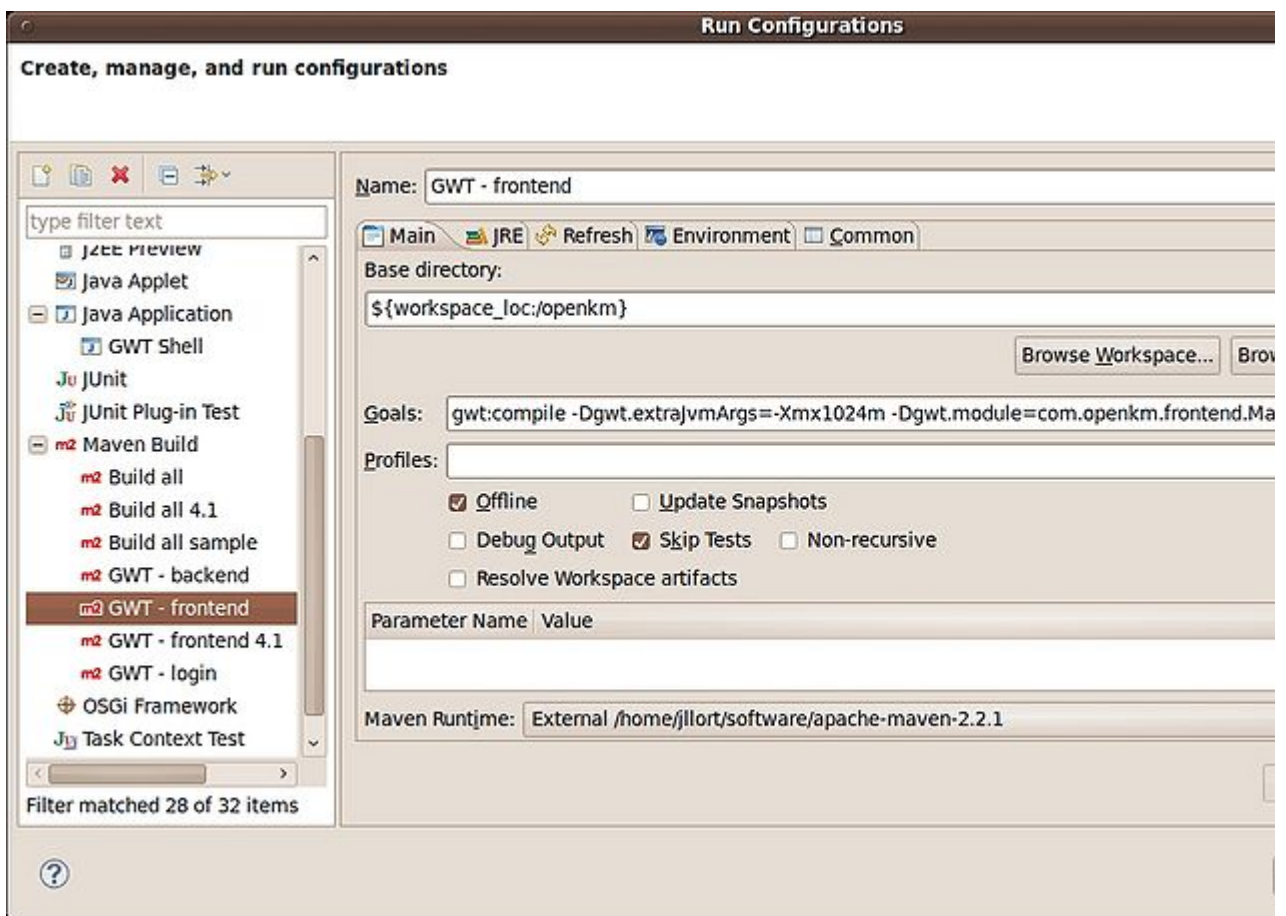
Compile only GWT frontend module

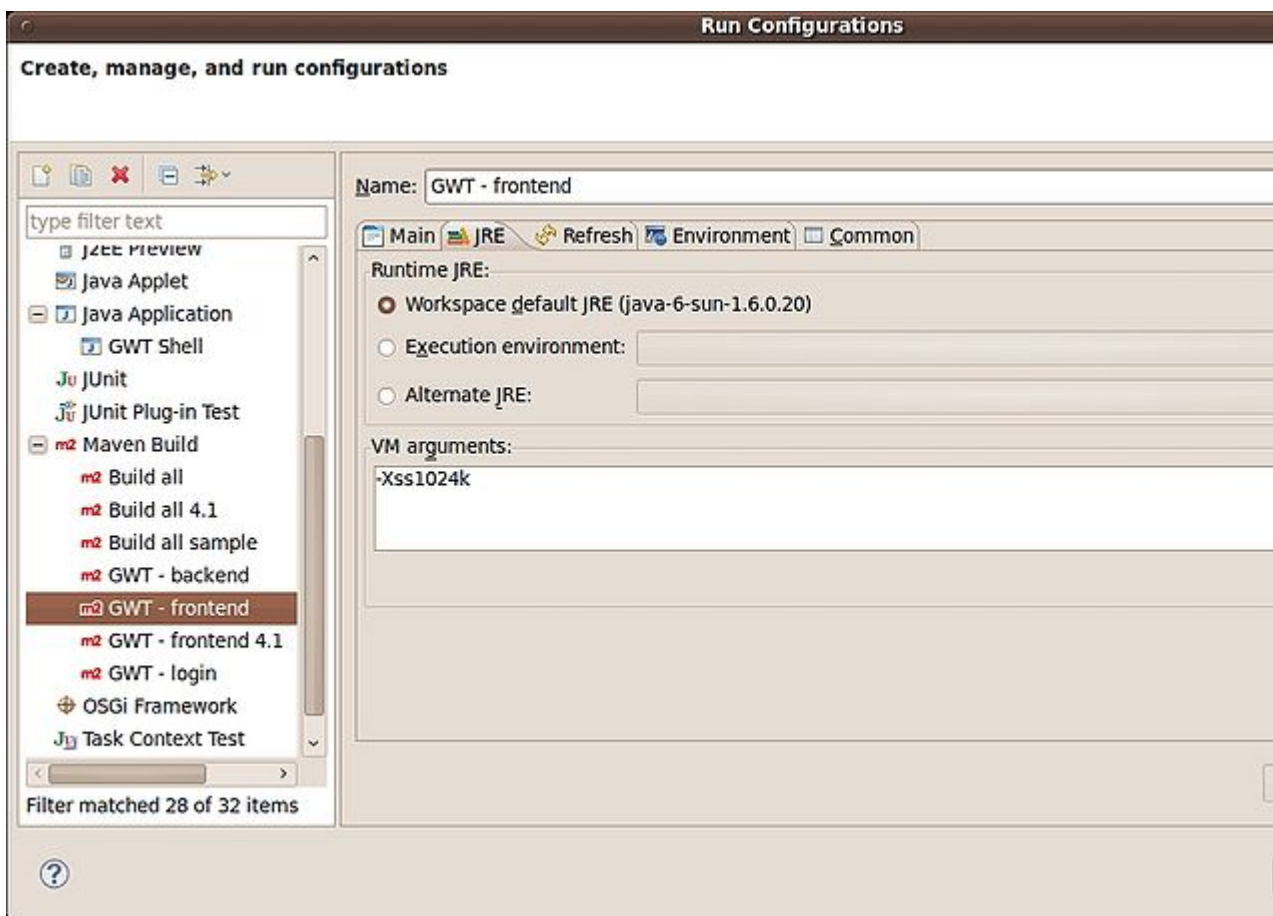
To compile GWT modules individually execute the command:

```
$ mvn gwt:compile -Dgwt.module=com.openkm.frontend.Main
```

Configure GWT build from eclipse

Go to "**Run configurations**", choose "**Maven build**" and click on "**right mouse button**" to "**add**" new maven build.





Compile only one GWT frontend permutation

GWT by default compiles six permutations, each one for distinct browsers. Each permutation consumes a lot of hardware resources (RAM and CPU's).



For fast development and less compilation time enable only a single permutation.

Decreasing the number of enabled permutation, you will decrease the total compilation time needed.

You can also declare several permutations at same time, for example for enabling firefox and chrome can use:

```
<set-property name="user.agent" value="gecko1_8, safari"/>
```

Enable only firefox permutation

Edit the main.gwt.xml OpenKM project file:

```
<!-- Compile for Firefox only -->
<set-property name="user.agent" value="gecko1_8"/>
```

Enable only chrome permutation

Edit the `main.gwt.xml` OpenKM project file:

```
<!-- Compile for Firefox only -->
<set-property name="user.agent" value="safari"/>
```

Enable only IE permutation

Edit the `main.gwt.xml` OpenKM project file:

```
<!-- Compile for Firefox only -->
<set-property name="user.agent" value="ie8"/>
```



Depending the GWT version also are allowed `ie6`, `ie9` and `ie10` values.

More information at [UserAgent.gwt.xml](#).

Configure number of CPU's to be used for compiling GWT frontend



Increasing the number of CPU's used by compiling GWT you will decrease the total compilation time needed.

GWT compilation process can use more than one cpu. You can set the number of cpu to be used with "**localworkers**" parameter.



Important, depending on the number of cores the memory decrease the number of gwt instances compiled at the same time, decreasing `localWorkers`, otherwise your computer may hang.

Edit the `pom.xml` project file.

```
<plugins>
  <plugin>
    <groupId>org.codehaus.mojo</groupId>
    <artifactId>gwt-maven-plugin</artifactId>
    <version>${gwt.version}</version>
    <executions>
      <execution>
        <goals>
          <goal>compile</goal>
          <!-- <goal>generateAsync</goal> -->
          <!-- <goal>test</goal> -->
        </goals>
      </execution>
    </executions>
    <configuration>
      <runTarget>com.openkm.frontend.Main/index.html</runTarget>
      <modules>
        <module>com.openkm.frontend.Main</module>
      </modules>
      <localWorkers>4</localWorkers>
    </configuration>
  </plugin>
```



*In case you're making a lot of changes in OpenKM UI (GWT) we recommend disabling the "**compile**" directive*

Deploying script sample

You can use a deploying script:

```
#!/bin/bash
# @author: Kenneth Walter

#Cleans and re-compiles the source files
mvn -Dmaven.test.skip=true clean gwt:compile install $*

#Set the TARGET_DIRECTORY to the path of your tomcat installation
TARGET_DIRECTORY=~/.Downloads/tomcat-7.0.27/
FILE_TO_MOVE=target/OpenKM.war
#This will only attempt to replace the existing WAR if the new WAR exists
if [ -f $FILE_TO_MOVE ]
then
    echo 'Deploying WAR to Tomcat Directory'
    cp -v $FILE_TO_MOVE $TARGET_DIRECTORY/webapps/.
    echo 'Done'
fi
```

Profiling application

You can profile local and remote Java applications using [Java VisualVM](#).



For the local application is easy because it detects automatically and show them in the listing. For the remote application you need to do a little work.

Edit Tomcat startup scripts

Edit the `$TOMCAT_HOME/bin/setenv.*` files (`setenv.sh` for linux and `setenv.bat` for Windows).



The `java.rmi.server.hostname` should match the IP of the server, usually the `127.0.0.1` value by the right IP.

```
CATALINA_OPTS="$CATALINA_OPTS -Dcom.sun.management.jmxremote=true"
CATALINA_OPTS="$CATALINA_OPTS -Dcom.sun.management.jmxremote.port=9090"
CATALINA_OPTS="$CATALINA_OPTS -Dcom.sun.management.jmxremote.ssl=false"
CATALINA_OPTS="$CATALINA_OPTS -Dcom.sun.management.jmxremote.authenticate=false"
CATALINA_OPTS="$CATALINA_OPTS -Djava.rmi.server.hostname=127.0.0.1"
```



To connect to this remote Tomcat instance:

- Go to **File > Add JMX Connection...** and fill the hostname and port.
- Additionally can change the port is 9090 using the `com.sun.management.jmxremote.port` property.

Authentication

To enhance security, enable authentication. Edit the `$TOMCAT_HOME/bin/setenv.*` files (`setenv.sh` for linux and `setenv.bat` for Windows).

```
CATALINA_OPTS="$CATALINA_OPTS -Dcom.sun.management.jmxremote.authenticate=true"
CATALINA_OPTS="$CATALINA_OPTS -Dcom.sun.management.jmxremote.password.file=$CATALINA_
CATALINA_OPTS="$CATALINA_OPTS -Dcom.sun.management.jmxremote.access.file=$CATALINA_HO
```

Edit the access allow file `$CATALINA_HOME/conf/jmxremote.access`:

```
monitorRole readonly
controlRole readwrite
```



To show remote threads you need the **readwrite** permission.

Edit the password file `$CATALINA_BASE/conf/jmxremote.password`:

```
monitorRole tomcat
controlRole tomcat
```

Restrict access to these files:

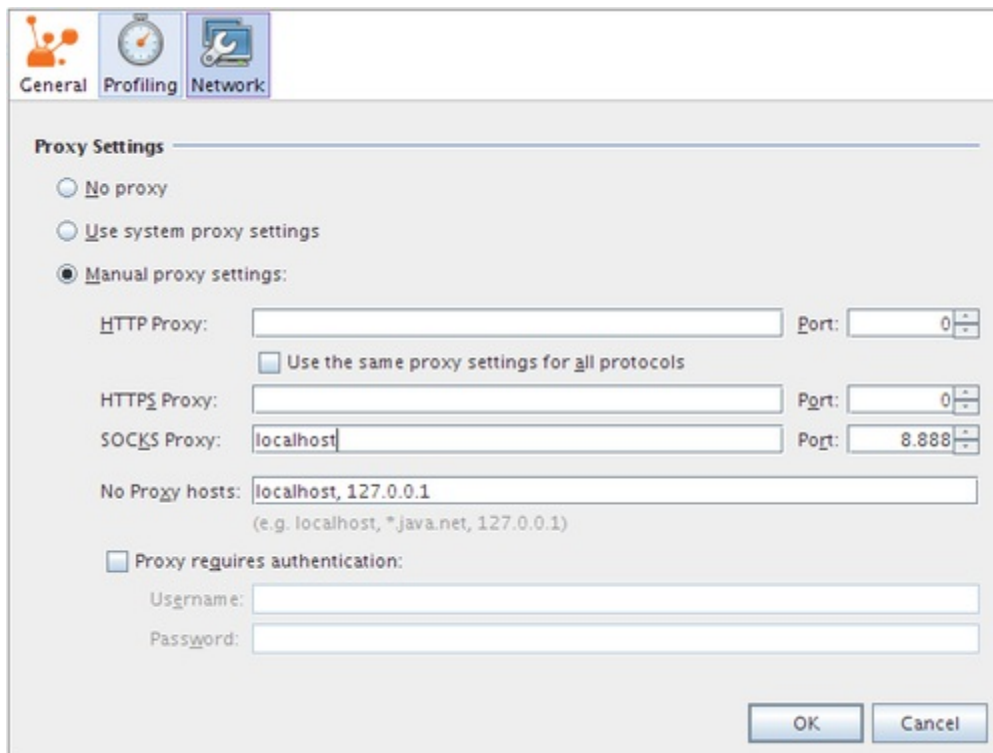
```
$ chmod 600 $CATALINA_HOME/conf/jmxremote.access
$ chmod 600 $CATALINA_HOME/conf/jmxremote.password
```

Monitoring Remote JVM Over SSH

The easiest way to connect to the remote JMX is to use a [SOCKS proxy](#).

```
$ ssh user@external.server.ip -D 8888
```

Configure VisualVM to use this proxy. Go to **Tools > Options > Network** and edit **Manual Proxy Settings**. Configure SOCKS Proxy at localhost and port 8888.



Now you can connect VisualVM to the remote target. Go to **File > Add JMX Connection...** and type the IP or hostname of the remote machine and the configured JMX port (in our sample configuration is 9090).

Connection:

Usage: <hostname>:<port> OR service:jmx:<protocol>:<sap>

☒ Display name:

☐ Use security credentials

Username:

Password:

☐ Save security credentials

OK Cancel



More information at:

- [Profiling With VisualVM, Part 1](#)
- [Profiling With VisualVM, Part 2](#)
- [Connecting Visual VM to Tomcat 7](#)
- [Monitoring and Managing Tomcat 7](#)
- [Remote Profiling of JBoss using VisualVM](#)
- [Troubleshooting application performance with VisualVM](#)
- [Monitoring of Tomcat with VisualVM and VisualGC](#)
- [Using VisualVM to fix live Tomcat and JVM problems](#)

HPROF

Java includes HPROF, a profiler which collect application runtime information. HPROF is capable of presenting CPU usage, heap allocation statistics, and monitor contention profiles.

For example, can collect CPU usage information by sampling threads.

Edit `$TOMCAT_HOME/bin/setenv.sh` file (or `setenv.bat` if using Windows):

```
CATALINA_OPTS="$CATALINA_OPTS -agentlib:hprof=cpu=samples"
```



When Tomcat starts you can see a file called **java.hprof.txt**. The CPU profiling info will be dumped to this file once Tomcat process is stopped.

More information at:

- [HPROF: A Heap/CPU Profiling Tool in J2SE 5.0](#)
- [HPjmeter](#)

Application version compatibility table



Application compatibility changes can be caused by:

- DTD definition changes. Usually newer application version supports older DTD versions.
- Libraries upgrades.



If your OpenKM server can't access to the Internet, you can download this DTD and copy to a conveniente place at you server. Remember to update your forms.xml with the TDT location.

```
<!DOCTYPE workflow-forms PUBLIC "-//OpenKM//DTD Workflow Forms 2.3//EN"
    "file:///path/to/workflow-forms-2.3.dtd">
```

Compatibility tables:

Property Groups	OpenKM
property-groups-2.3.dtd	Since 6.3.1
property-groups-2.2.dtd	Since 6.3.0

Workflow Forms	OpenKM
workflow-forms-2.5.dtd	Since 6.3.1
workflow-forms-2.4.dtd	Since 6.3.0

Report Parameters
report-parameters-2.0.dtd

Jasper Reports Engine	OpenKM
3.7.6	Since 6.3.0

Creating a JAR file with Eclipse



These instructions are only a guide for creating a JAR based on a single .java file.

Take it as is, an introduction.

The following steps can be changed depending on the new Eclipse version changes.

1. **Choose the Java file** from Project Explorer or Navigator view.
2. **Do right click** (will be shown contextual menu) and choose "export".
3. Choose **Java -> JAR file**.
4. Select the **destination folder**, for example C:\openkm-dev\jar\your-class.jar.
5. **Click Next**.
6. **Click Next**.
7. **Optional step**. If your java class implements "**public static void main(String[] args)**" then you must choose your java class as the class of the entry point.
8. **Click Finish**.

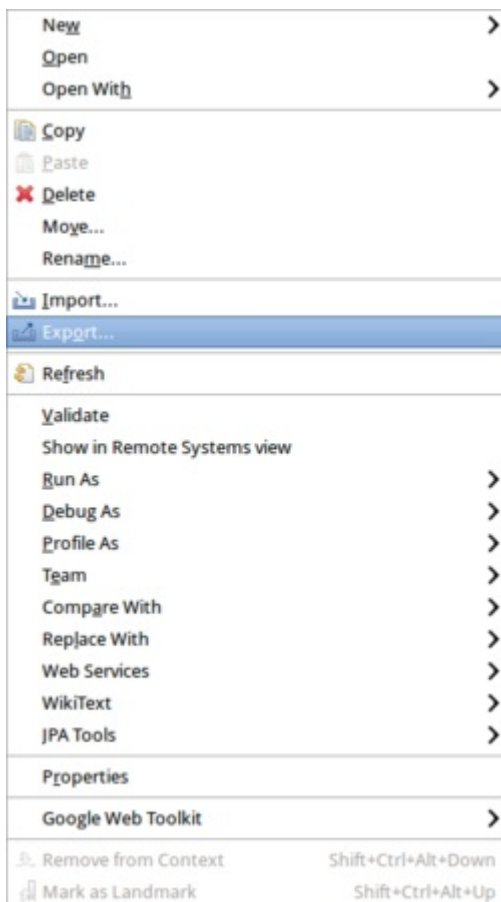


Step 7 for example is needed when you're creating a new crontab job.

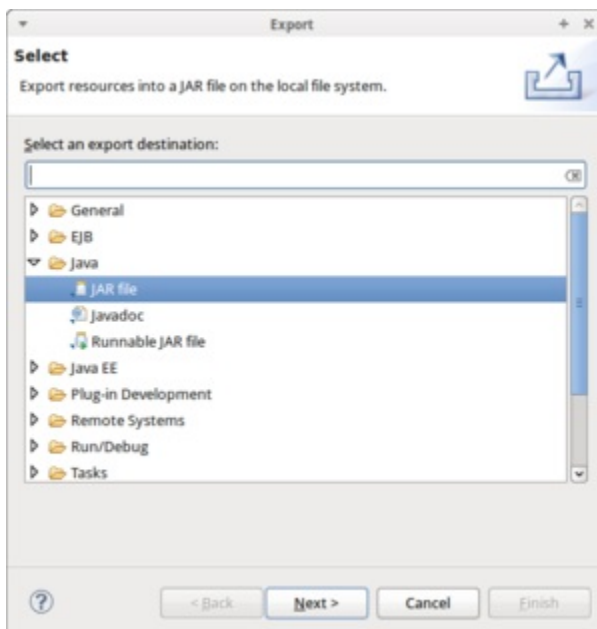
More information at [Creating your own Crontab job](#).

Images

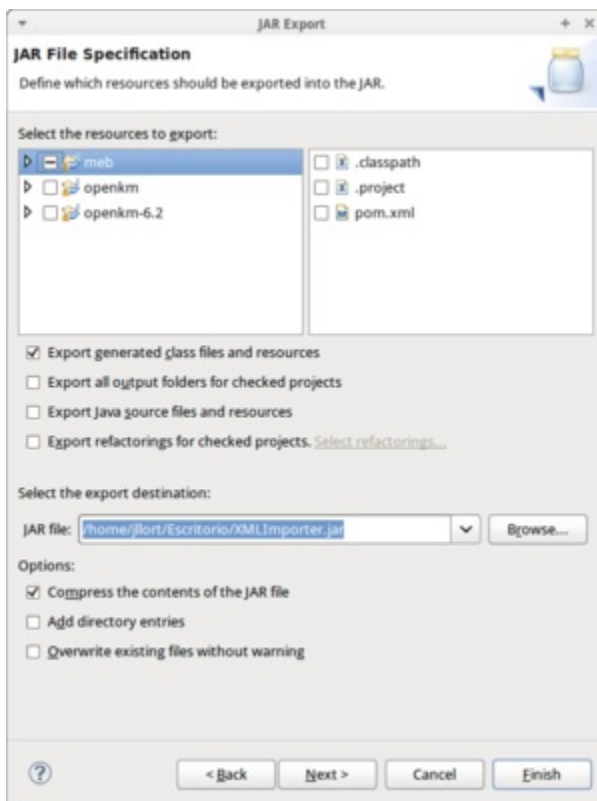
Select the file to be exported as a jar. Right click on mouse and select "**export**" option.



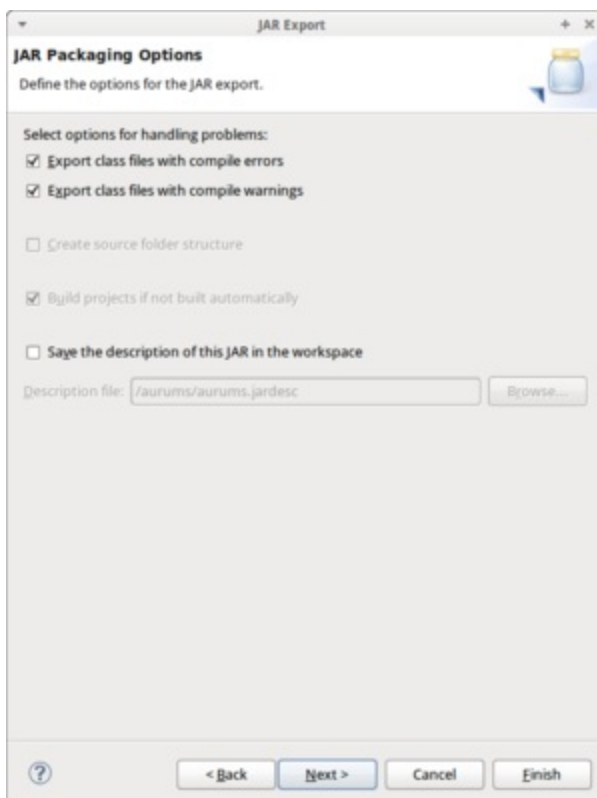
Select "**jar**" file option and click on "**next**" button.



Select the "**JAVA classes to be included**" into the jar file and click on "**next**" button.



Click on "**next**" button.



Select the "**Main class**" and click on "**Finish**" button.



If your JAVA class implements "**public static void main(String[] args)**" then you must choose your java Main class as the class of the entry point, otherwise you can skip this step.

JAR Manifest Specification
Customize the manifest file for the JAR file.

Specify the manifest:

☒ Generate the manifest file

☐ Save the manifest in the workspace

☐ Use the saved manifest in the generated JAR description file

Manifest file:

☐ Use existing manifest from workspace

Manifest file:

Seal contents:

☐ Seal the JAR

☒ Seal some packages

Nothing sealed

Select the class of the application entry point:

Main class:

Creating plugins

OpenKM plugin system helps you to add software components increasing the application features.




*OpenKM plugin system is based on **hot deploy** of JAR packages.*

From OpenKM plugin system you can

- *Register new Automation actions.*
- *Register new Automation validations.*

Register a new plugin

- To install new plugin **create a jar file** and **copy into your \$TOMCAT/plugins** folder. More information at [Creating a JAR file with Eclipse](#).
- Go to **Administration > Utilities > Plugins** and click at top right the  Reload plugins button.
- In **plugins table** will be shown the new plugin.



A plugin configuration needs to change [Configuration parameters](#) or restart application. For example [Creating your own Automation Action](#) case.

Creating your own Automation Action



Plugin architecture is available from OpenKM version 6.3.7 for older versions take a look at [Creating your own Automation Action \(deprecated \)](#).

You can create your own Automation Action.

Conditions:

- The new Automation Action class **must implement the "Action" interface**.
- The new Automation Action class **must be declared under the package "com.openkm.automation.action"**.
- The new Automation Action class **must be annotated with "@PluginImplementation"**.

Validation interface:

```
package com.openkm.automation;

import java.util.Map;

import net.xeoh.plugins.base.Plugin;

public interface Action extends Plugin {
    public static final String METHOD_PRE = "executePre";
    public static final String METHOD_POST = "executePost";





    public void executePre(Map<String, Object> env, Object... params);
    public void executePost(Map<String, Object> env, Object... params);
    public boolean hasPost();
    public boolean hasPre();
    public String getName();
    public String getParamType00();
    public String getParamSrc00();
    public String getParamDesc00();
    public String getParamType01();
    public String getParamSrc01();
    public String getParamDesc01();
    public String getParamType02();
    public String getParamSrc02();
    public String getParamDesc02();
}
```



More information about [Register a new plugin](#).

Methods description

Method	Type	Description
<code>executePre(Map<String, Object> env, Object... params)</code>	void	The method executed by the Automation event when the validation conditions succeed, on "pre" stage.

<i>executePost(Map<String, Object> env, Object... params)</i>	void	The method executed by the Automation event when the validation conditions succeed, on "post" stage.
<i>hasPost()</i>	boolean	<p>When return true, indicates the executePost(...) method can be used in the Automation post stage.</p> <div>  More information about automation stages at Automation. </div>
<i>hasPre()</i>	boolean	<p>When returns true, indicates the executePre(...) method can be used in the Automation pre stage.</p> <div>  More information about automation stages can be found at Automation. </div>
<i>getName()</i>	String	Sets the name that will be shown in the administrator user interface selector list.
<i>getParamType00</i> <i>getParamType01</i> <i>getParamType02</i>	String	<p>Sets the parameter type.</p> <p>Available values:</p> <ul style="list-style-type: none"> Automation.PARAM_TYPE_EMPTY Automation.PARAM_TYPE_TEXT Automation.PARAM_TYPE_INTEGER Automation.PARAM_TYPE_BOOLEAN Automation.PARAM_TYPE_TEXTAREA Automation.PARAM_TYPE_CODE Automation.PARAM_TYPE_USER Automation.PARAM_TYPE_ROLE Automation.PARAM_TYPE_OMR <div>  When value is set to Automation.PARAM_TYPE_EMPTY </div> <div>  The parameters work as a logical group. </div>

		[<i>getParamType00</i> + <i>getParamSrc00</i> + <i>getParamDesc00</i>]
<i>getParamSrc00</i> <i>getParamSrc01</i> <i>getParamSrc02</i>	String	Sets the source type. Available values: <ul style="list-style-type: none"> • <i>Automation.PARAM_SOURCE_EMPTY</i> • <i>Automation.PARAM_SOURCE_FOLDER</i> • <i>Automation.PARAM_SOURCE_OMR</i>
<i>getParamDesc00</i> <i>getParamDesc01</i> <i>getParamDesc02</i>	String	The parameter description.

Understanding env variable

The env variable is present on *isValid* method, it is a Map of values injected by automation. These map values provide information about the node involved on the event and other related information.



It takes some time, until you take control of the AutomationUtils class, because they are centralized, the methods to retrieve data.

For example to retrieve the uuid of the node that caused the event.

```
String uuid = AutomationUtils.getUuid(env);
```

Understanding params variable

The env variable is present on *isValid* method, it is an array of Objects. This array is filled by Automation event based on the number of parameters set on Action class (which will not be *Automation.PARAM_TYPE_EMPTY* value).

The array can have several Objects with a distinct type, it is necessary retrieve the parameters to the correct object type.



It takes some time, until you take control of AutomationUtils class, because they are centralized, the methods to retrieve parameters.

For example to retrieve the first parameters as String object.

```
String param00 = AutomationUtils.getString(0, params);
```

Description of available ParamType values:

<i>Property</i>	<i>Description</i>
<i>Automation.PARAM_TYPE_EMPTY</i>	<i>Indicates an empty value.</i>
<i>Automation.PARAM_TYPE_TEXT</i>	<i>Indicates a text value will be required.</i>
<i>Automation.PARAM_TYPE_INTEGER</i>	<i>Indicates an integer value will be required.</i>
<i>Automation.PARAM_TYPE_BOOLEAN</i>	<i>Indicates a boolean value will be required.</i>
<i>Automation.PARAM_TYPE_TEXTAREA</i>	<i>Indicates a text area value will be required.</i>
<i>Automation.PARAM_TYPE_CODE</i>	<i>Indicates that code will be required.</i>
<i>Automation.PARAM_TYPE_USER</i>	<i>Indicates that a valid application user will be required.</i>
<i>Automation.PARAM_TYPE_ROLE</i>	<i>Indicates that a valid application role will be required.</i>
<i>Automation.PARAM_TYPE_OMR</i>	<i>Indicates that a valid OMR id - object mark recognition - will be required.</i>

Description of available Src values:

<i>Property</i>	<i>Description</i>
<i>PARAM_SOURCE_EMPTY</i>	<i>Indicates an empty source.</i>
<i>PARAM_SOURCE_FOLDER</i>	<i>Indicates that the source must be a folder.</i>
<i>PARAM_SOURCE_OMR</i>	<i>Indicates that the source must be a valid OMR - object mark recognition -.</i>

Detecting recursion

When you create your automation actions, should consider recursion case. For example if your action goes linked with

"CREATE_DOCUMENT" event and into the action you create a new document you can go into an infinite loop. For it when you create a new Action should take special care of this cases.



Basic recursive samples:

- Automation task linked with **"Create document"** event when into is set an **action** that also **creates a new document**.
- Automation task linked with **"Set metadata group"** event when into is set an **action** that also **changes metadata group values**.

Complex recursive sample:

- Automation task linked with **"Create document"** event when into is set an action that **changes metadata group values** and automation task linked with **"Set metadata group"** event when into is set an action that **creates a new document**.

Code to detect recursive calls caused by automation action classes:

```
if (StackTraceUtils.isCallingMe(this.getClass().getName())) {
    log.info("Recursion detected");
    return;
}
```

Example

AddKeyword class:

```
package com.openkm.automation.action;

import java.util.Map;

import com.openkm.automation.Action;
import com.openkm.automation.AutomationUtils;
import com.openkm.core.Config;
import com.openkm.dao.NodeBaseDAO;
import com.openkm.dao.bean.Automation;

import net.xeoh.plugins.base.annotations.PluginImplementation;

/**
 * AddKeyword
 */
@PluginImplementation
public class AddKeyword implements Action {

    @Override
    public void executePre(Map<String, Object> env, Object... params) throws Exception {
    }

    @Override
    public void executePost(Map<String, Object> env, Object... params) throws Exception {
        String keyword = AutomationUtils.getString(0, params);
    }
}
```

```
String uuid = AutomationUtils.getUuid(env);

if (uuid != null && keyword != null && !keyword.isEmpty()) {
    if (Config.SYSTEM_KEYWORD_LOWERCASE) {
        keyword = keyword.toLowerCase();
    }

    NodeBaseDAO.getInstance().addKeyword(uuid, keyword);
}

@Override
public boolean hasPost() {
    return true;
}

@Override
public boolean hasPre() {
    return false;
}

@Override
public String getName() {
    return "AddKeyword";
}

@Override
public String getParamType00() {
    return Automation.PARAM_TYPE_TEXT;
}

@Override
public String getParamSrc00() {
    return Automation.PARAM_SOURCE_EMPTY;
}

@Override
public String getParamDesc00() {
    return "Keyword";
}

@Override
public String getParamType01() {
    return Automation.PARAM_TYPE_EMPTY;
}

@Override
public String getParamSrc01() {
    return Automation.PARAM_SOURCE_EMPTY;
}

@Override
public String getParamDesc01() {
    return "";
}

@Override
public String getParamType02() {
    return Automation.PARAM_TYPE_EMPTY;
}

@Override
public String getParamSrc02() {
    return Automation.PARAM_SOURCE_EMPTY;
}

@Override
```

```
public String getParamDesc02() {  
    return "";  
}
```

Creating your own Automation Validation



Plugin architecture is available from OpenKM version 6.3.7 for older versions take a look at [Creating your own Automation Validation \(deprecated \)](#).

You can create your own Automation Validation.

Conditions:

- The new Automation Validation class **must implement the "Validation" interface**.
- The new Automation Validation class **must be declared under the package "com.openkm.automation.validation"**.
- The new Automation Validation class **must be annotated with "@PluginImplementation"**.

Validation interface:

```
package com.openkm.automation;

import java.util.Map;

import net.xeoh.plugins.base.Plugin;

public interface Validation extends Plugin {
    public static final String METHOD = "isValid";

    boolean isValid(Map<String, Object> env, Object... params);
    public boolean hasPost();
    public boolean hasPre();
    public String getName();
    public String getParamType00();
    public String getParamSrc00();
    public String getParamDesc00();
    public String getParamType01();
    public String getParamSrc01();
    public String getParamDesc01();
    public String getParamType02();
    public String getParamSrc02();
    public String getParamDesc02();
}
```



More information about [Register a new plugin](#).

Methods description

Method	Type	Description
isValid(Map<String, Object> env, Object... params)	boolean	The method executed by the Automation event to evaluate if conditions are true or false.

hasPost()	boolean	<p>When returns true, indicates the isValid(...) method can be used in Automation post stage.</p> <div>  More information about automation stages at Automation. </div>
hasPre()	boolean	<p>When returns true, indicates the isValid(...) method can be used in Automation pre stage.</p> <div>  More information about automation stages at Automation. </div>
getName()	String	Sets the name that will be shown in the administrator user interface selector list.
getParamType00 getParamType01 getParamType02	String	<p>Sets the parameter type.</p> <p>Available values:</p> <ul style="list-style-type: none"> Automation.PARAM_TYPE_EMPTY Automation.PARAM_TYPE_TEXT Automation.PARAM_TYPE_INTEGER Automation.PARAM_TYPE_BOOLEAN Automation.PARAM_TYPE_TEXTAREA Automation.PARAM_TYPE_CODE Automation.PARAM_TYPE_USER Automation.PARAM_TYPE_ROLE Automation.PARAM_TYPE_OMR <div>  When the value is set to Automation.PARAM_TYPE_EMPTY </div> <div>  The parameters work as a logical group. [getParamType00 + getParamSrc00 + getParamDesc00] </div>
getParamSrc00	String	Set the source type.

<i>getParamSrc01</i>		Available values:
<i>getParamSrc02</i>		
		<ul style="list-style-type: none"> • <i>Automation.PARAM_SOURCE_EMPTY</i> • <i>Automation.PARAM_SOURCE_FOLDER</i> • <i>Automation.PARAM_SOURCE_OMR</i>
<i>getParamDesc00</i>	String	The parameter description.
<i>getParamDesc00</i>		
<i>getParamDesc00</i>		

Understanding env variable

The env variable is present on *isValid* method, it is a Map of values injected by automation. These map values provide information about the node involved on event and other related information.



It takes some time, until you take control of *AutomationUtils* class, because they are centralized, the methods for retrieve data.

For example to retrieve the uuid of the node that caused the event.

```
String uuid = AutomationUtils.getUuid(env);
```

Understanding params variable

The env variable is present on *isValid* method, it is an array of Objects. This array is filled by Automation event based on the number of parameters set on Validation class (which will not be *Automation.PARAM_TYPE_EMPTY* value).

The array can have several Objects with a distinct type, it is necessary to retrieve the parameters to the correct object type.



It takes some time, until you take control of *AutomationUtils* class, because they are centralized, the methods to retrieve parameters.

For example to retrieve the first parameters as String object.

```
String param00 = AutomationUtils.getString(0, params);
```

Description of available ParamType values:

Property	Description

<i>Automation.PARAM_TYPE_EMPTY</i>	<i>Indicates an empty value.</i>
<i>Automation.PARAM_TYPE_TEXT</i>	<i>Indicates that a text value will be required.</i>
<i>Automation.PARAM_TYPE_INTEGER</i>	<i>Indicates that an integer value will be required.</i>
<i>Automation.PARAM_TYPE_BOOLEAN</i>	<i>Indicates that a boolean value will be required.</i>
<i>Automation.PARAM_TYPE_TEXTAREA</i>	<i>Indicates that a text area value will be required.</i>
<i>Automation.PARAM_TYPE_CODE</i>	<i>Indicates that a code will be required.</i>
<i>Automation.PARAM_TYPE_USER</i>	<i>Indicates that a valid application user will be required.</i>
<i>Automation.PARAM_TYPE_ROLE</i>	<i>Indicates that a valid application role will be required.</i>
<i>Automation.PARAM_TYPE_OMR</i>	<i>Indicates that a valid OMR id - object mark recognition - will be required.</i>

Description of available Src values:

<i>Property</i>	<i>Description</i>
<i>PARAM_SOURCE_EMPTY</i>	<i>Indicates an empty source.</i>
<i>PARAM_SOURCE_FOLDER</i>	<i>Indicates that the source must be a folder.</i>
<i>PARAM_SOURCE_OMR</i>	<i>Indicates that the source must be a valid OMR - object mark recognition -.</i>

Example

PathContains class:

```
package com.openkm.automation.validation;

import java.util.Map;

import com.openkm.api.OKMRepository;
```

```
import com.openkm.automation.AutomationUtils;
import com.openkm.automation.Validation;
import com.openkm.dao.bean.Automation;

import net.xeoh.plugins.base.annotations.PluginImplementation;

/**
 * Check if the current parent path contains a designed one. The only
 * parameter is a path and will test if this one is included in the
 * actual parent.
 */
@PluginImplementation
public class PathContains implements Validation {

    @Override
    public boolean isValid(Map<String, Object> env, Object... params) throws Exception {
        String fldUuid = AutomationUtils.getString(0, params);
        String parentPath = AutomationUtils.getParentPath(env);
        String fldPath = OKMRepository.getInstance().getNodePath(null, fldUuid);
        if (parentPath.startsWith(fldPath)) {
            return true;
        } else {
            return false;
        }
    }

    @Override
    public boolean hasPost() {
        return true;
    }

    @Override
    public boolean hasPre() {
        return true;
    }

    @Override
    public String getName() {
        return "PathContains";
    }

    @Override
    public String getParamType00() {
        return Automation.PARAM_TYPE_TEXT;
    }

    @Override
    public String getParamSrc00() {
        return Automation.PARAM_SOURCE_FOLDER;
    }

    @Override
    public String getParamDesc00() {
        return "String";
    }

    @Override
    public String getParamType01() {
        return Automation.PARAM_TYPE_EMPTY;
    }

    @Override
    public String getParamSrc01() {
        return Automation.PARAM_SOURCE_EMPTY;
    }
}
```

```
@Override
public String getParamDesc01() {
    return "";
}

@Override
public String getParamType02() {
    return Automation.PARAM_TYPE_EMPTY;
}

@Override
public String getParamSrc02() {
    return Automation.PARAM_SOURCE_EMPTY;
}

@Override
public String getParamDesc02() {
    return "";
}
}
```

Creating templates

You can create new documents based on Templates. The application can handle 3 types of template documents:

- PDF
- OpenOffice
- Text (This includes other text-based like HTML, for example).

In order to generate the final document from the template, we need to make an association with [Metadata](#). The final document will be assigned the corresponding [Metadata](#) with the user input values, so you can use them for a future search.

The template syntax, which is used by the [FreeMarker Template Engine](#). This engine can be used from making simply variable substitution to loops and other control structure.



In the samples we use only the basic ones, but you can learn more at [FreeMarker Manual](#).

Creating OpenOffice and LibreOffice templates

Preliminaries

Register the Metadata group definition at **Administration** > [Metadata](#).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.3//EN"
    "http://www.openkm.com/dtd/property-groups-2.3.dtd">

<property-groups>
  <property-group label="Sample Template" name="okg:tpl">
    <input label="Name" name="okp:tpl.name"/>
    <input label="Birth Date" name="okp:tpl.birth_date" type="date" />
    <select label="Language" name="okp:tpl.language" type="simple">
      <option label="Java" value="java"/>
      <option label="Python" value="python"/>
      <option label="PHP" value="php" />
    </select>
  </property-group>
</property-groups>
```

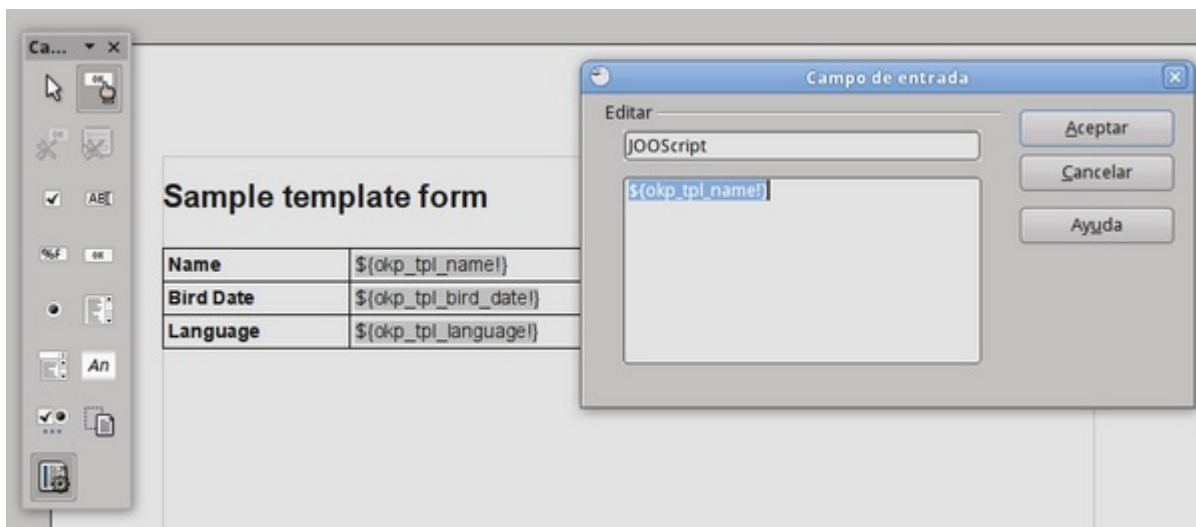
In order to make a relation between the template form field and the property group, you have to name them following a naming convention: replace ":" and "." by "_". Here you can see the matching between both fields:

Metadata field name	Template field name	Template field value
okp:tpl.name	okp_tpl_name	\${okp_tpl_name!}
okp:tpl.birth_date	okp_tpl_birth_date	\${okp_tpl_birth_date!}
okp:tpl.language	okp_tpl_language	\${okp_tpl_language!}

Creating the template

To create a template in OpenOffice.org go to **View** > **Toolbars** > **Form Controls**.

The a new dialogue box will appear and you will be able to add form elements to the document. This dialogue box contains stuff such as Labels, Text Boxes, Check Boxes, etc.



The field that will be replaced with the value of a variable passed can be specified using the standard FreeMarker expression notation by typing the variable name into the document as follows:

```
Hi ${name}!
```

However, mixing processing instructions and normal text into the document may become confusing and clutter the layout. So JODReports provides an alternative way of inserting fields. You can insert a visual field in OpenOffice.org Writer from the menu **Insert / Fields / Other...** (or just press **Ctrl+F2**), then click on the **Functions** tab and select **Input field**. Change the field Reference to **JOOScript**. In the following field Content enter the variable name (e.g. "\${name}"). This field will typically be displayed by Writer as a grayed rectangle with the variable name, while moving the mouse over it will reveal the field reference.

This is the OpenOffice.org document used in this sample [tpl.odt](#).

Read about OpenOffice.org templates at [JODReports](#).

Date format

JODReports uses FreeMarker inside, so you can take advantage of the FreeMarker formatting flexibilities. For example you can set date format by this way:

```
${okp_tpl_birth_date?string("yyyy-MM-dd")}
```

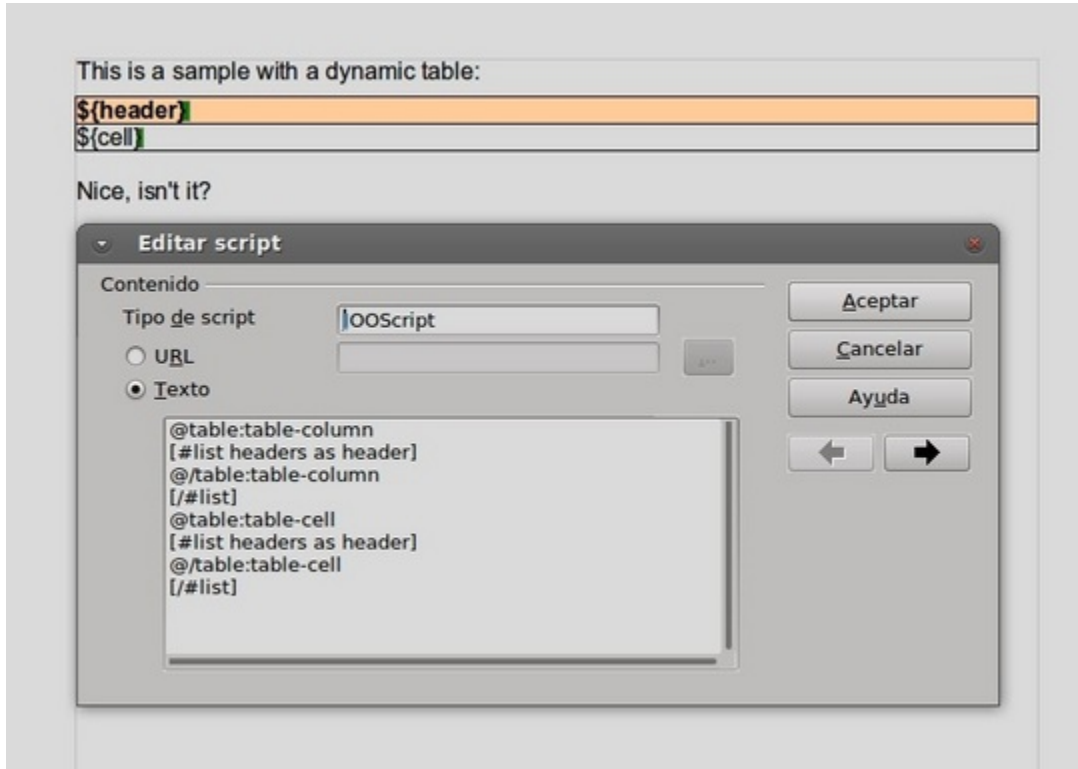
If date may be null, then you need to check this:

```
<#if okp_tpl_birth_date??>${okp_tpl_birth_date?string("yyyy-MM-dd")}</#if>
```

Learn more about FreeMarker formatting at [Built-in Reference](#).

Dynamic tables

You can even use the template feature to generate dynamic tables. Lets see an example.



In the screen shot you can see a document with a table. The header has a colored background and also is bold. The script shows the options to generate the table headers.

This is the OpenOffice.org document used in this sample: [dynamic_table.odt](#)

Learn more about this feature at:

- [JODReports: Inserting Directives at Special Locations.](#)
- [Dynamic table column count](#)

Creating PDF template

You can create PDF template forms with several applications:

- [Acrobat Pro](#)
- [How-to: Create PDF forms with OpenOffice](#)
- [PDF Forms Designer](#)



Every PDF form field name needs to be unique, you can't set the same property value to several form fields.

When converting from OpenOffice document to PDF, repeated form fields are renamed to `original_name_2`, `original_name_3`, etc.

If you need to put the same property value in several fields, try [Creating OpenOffice and LibreOffice templates](#).



In the sample below, we've used OpenOffice to create PDF template.

Preliminaries

Register the Metadata group definition at **Administration** > [Metadata](#).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.1//EN"
    "http://www.openkm.com/dtd/property-groups-2.1.dtd">
<property-groups>
  <property-group label="Sample Template" name="okg:tpl">
    <input label="Name" name="okp:tpl.name"/>
    <input label="Birth Date" name="okp:tpl.birth_date" type="date" />
    <select label="Language" name="okp:tpl.language" type="simple">
      <option label="Java" value="java"/>
      <option label="Python" value="python"/>
      <option label="PHP" value="php" />
    </select>
  </property-group>
</property-groups>
```

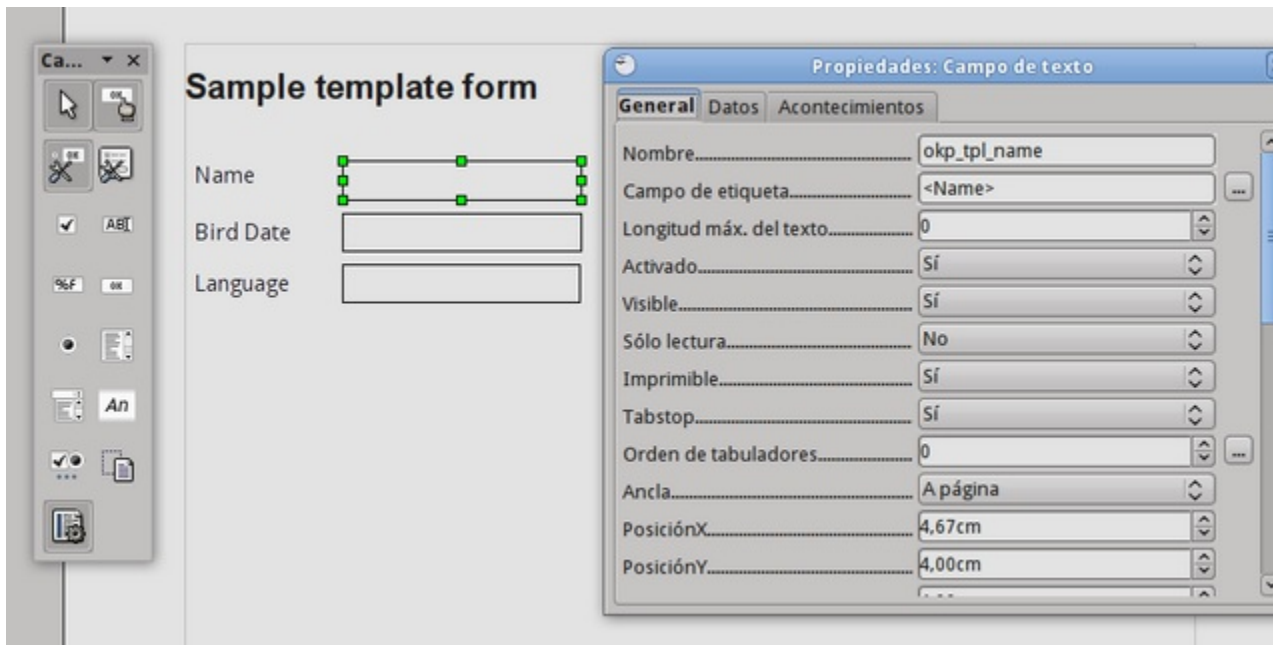
In order to make a relation between the template form field and the property group, you have to name them following a naming convention: replace ":" and "." by "_". Here you can see the matching between both fields:

Metadata field name	Template field name	Template field value
<code>okp:tpl.name</code>	<code>okp_tpl_name</code>	<code>\${okp_tpl_name!}</code>
<code>okp:tpl.birth_date</code>	<code>okp_tpl_birth_date</code>	<code>\${okp_tpl_birth_date!}</code>
<code>okp:tpl.language</code>	<code>okp_tpl_language</code>	<code>\${okp_tpl_language!}</code>

Creating the template

To create a template in OpenOffice.org go to **View > Toolbars > Form Controls**.

The a new dialogue will appear and you will be able to add form elements to the document. This dialogue contains stuff such as Labels, Text Boxes, Check Boxes, etc.



Once the form has been created, go to **File > Export as PDF** to generate a PDF with these form fields. This is the final result:

Sample template form

Name	<input type="text"/>
Bird Date	<input type="text"/>
Language	<input type="text"/>

Here you have the generated PDF file [tpl.pdf](#) and this is the ODT source [tpl_pdf.odt](#).

Date format

OpenKM uses *FreeMarker* inside, so you can take advantage of the *FreeMarker* formatting flexibilities. For example you can set date format by this way:

```
${okp_tpl_birth_date?string("yyyy-MM-dd")}
```

Learn more about *FreeMarker* formatting at [Built-in Reference](#).

Creating Text templates

Preliminaries

Register the Metadata group definition at **Administration** > [Metadata](#) .

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.1//EN"
"http://www.openkm.com/dtd/property-groups-2.1.dtd">

<property-groups>
  <property-group label="Sample Template" name="okg:tpl">
    <input label="Name" name="okp:tpl.name"/>
    <input label="Birth Date" name="okp:tpl.birth_date" type="date" />
    <select label="Language" name="okp:tpl.language" type="simple">
      <option label="Java" value="java"/>
      <option label="Python" value="python"/>
      <option label="PHP" value="php" />
    </select>
  </property-group>
</property-groups>
```

In order to make a relation between the template form field and the property group, you have to name them following a naming convention: replace ":" and "." by "_". Here you can see the matching between both fields:

Metadata field name	Template field name	Template field value
okp:tpl.name	okp_tpl_name	\${okp_tpl_name!}
okp:tpl.birth_date	okp_tpl_birth_date	\${okp_tpl_birth_date!}
okp:tpl.language	okp_tpl_language	\${okp_tpl_language!}

Creating the template

For this kind of template we will create an HTML one. This is the template source code:

```
<html>
  <body>
    <h1>Sample template</h1>
    <table>
      <tr>
        <td><b>Name</b></td>
        <td>${okp_tpl_name!}</td>
      </tr>
      <tr>
        <td><b>Birth Date</b></td>
        <td>${okp_tpl_birth_date!?string("yyyy-MM-dd")}</td>
      </tr>
      <tr>
        <td><b>Language</b></td>
```

```
<td>${okp_tpl_language!}</td>
</tr>
</table>
</body>
</html>
```

The tags `${okp_tpl_name!}`, `${okp_tpl_birth_date!}` and `${okp_tpl_language!}` will be replaced by the user input values. An error will occur and abort the template processing if you try to access a missing variable, but the `!` operator handles this situation. In addition, you can put a default value in case of a missing one with this expression:

```
${missing_value!"The default one"}
```

The result is a PDF conversion of the HTML template with the client input requested by OpenKM.

Date format

OpenKM uses FreeMarker inside, so you can take advantage of the FreeMarker formatting flexibilities. For example you can set date format by this way:

```
${okp_tpl_birth_date?string("yyyy-MM-dd")}
```

Learn more about FreeMarker formatting at [Built-in Reference](#).

Creating your own Automation Action (deprecated)



Automation architecture from version 6.3.7 has been changed instead of plugin architecture. This documentation section is for OpenKM 6.3.6 and older.

Users from version 6.3.7 should use this documentation section [Creating your own Automation Action](#).

You can create your own Automation Action.

Conditions:

- The new Automation Action class **must implement the "Action"** interface.
- The new Automation Action class **must be declared under the package "com.openkm.automation.action"**.



The new Automation Action class must be **within the OpenKM source code**. You might be interested in [OpenKM portable dev environment](#) for it.

Validation interface:

```
package com.openkm.automation;

import java.util.HashMap;

public interface Action {
    public static final String METHOD_PRE = "executePre";
    public static final String METHOD_POST = "executePost";

    public void executePre(HashMap<String, Object> env, Object... params);
    public void executePost(HashMap<String, Object> env, Object... params);
}
```

Methods description

Method	Type	Description
<code>executePre(Map<String, Object> env, Object... params)</code>	void	The method executed by the Automation event when the validation conditions succeed, on "pre" stage.
<code>executePost(Map<String, Object> env, Object... params)</code>	void	The method executed by the Automation event when the validation conditions succeed, on "post" stage.

Understanding env variable

The env variable is present on isValid method, it is a Map of values injected by automation. These map values provide information about the node involved on the event and other related information.



It takes some time, until you take control of the AutomationUtils class, because they are centralized, the methods to retrieve data.

For example to retrieve the uuid of the node that caused the event.

```
String uuid = AutomationUtils.getUuid(env);
```

Understanding params variable

The env variable is present on isValid method, it is an array of Objects. This array is filled by Automation event based on the number of parameters set on Validation class.

The array can have several Objects with a distinct type, it is necessary retrieve the parameters to the correct object type.



It takes some time, until you take control of AutomationUtils class, because they are centralized, the methods to retrieve parameters.

For example to retrieve the first parameters as String object.

```
String param00 = AutomationUtils.getString(0, params);
```

Preventing recursion

When you create your automation actions, should consider recursion case. For example if your action goes linked with "CREATE_DOCUMENT" event and into the action you create a new document you can go into an infinite loop. For it when you create a new Action should take special care of this cases.



Basic recursive samples:

- Automation task linked with "**Create document**" event when into is set an **action** that also **creates a new document**.
- Automation task linked with "**Set metadata group**" event when into is set an **action** that also **changes metadata group values**.


Complex recursive sample:





- Automation task linked with "**Create document**" event when into is set an action that **changes metadata group values** and automation task linked with "**Set metadata group**" event when into is set an action that **creates a new document**.

Register

Once you have created the new Automation Action, build the *OpenKM.war* file and deployed into the tomcat, it must be registered into the *OKM_AUTO_METADATA* table;

The *OKM_AUTO_METADATA* columns description:

Column	Description
AMD_ID	Autoincremental unique Id
AMD_AT	Indicates when the Automation Action can be executed, the values are: <ul style="list-style-type: none"> • "pre" indicate can be execute immediately before the action who has fired it. • "post" indicate can be execute after the action who has fired it.
AMD_CLASS_NAME	Java class name.
AMD_DESC00	Description of the parameter 0.
AMD_DESC01	Description of the parameter 1.
AMD_DESC02	Description of the parameter 2.
AMD_GROUP	Type of automation, allowed values are: <ul style="list-style-type: none"> • "action".
AMD_NAME	Name.
AMD_SRC00	Source, allowed values are: <ul style="list-style-type: none"> • "" default value. • "okm:folder" indicate the value is a folder uuid. <div>  When type "okm:folder" is applied will be shown a folder selector popup. </div>
AMD_SRC01	Source, allowed values are:

	<ul style="list-style-type: none"> • "" default value. • "okm:folder" indicate the value is a folder uuid. <div>  When type "okm:folder" is applied will be shown a folder selector popup. </div>
AMD_SRC02	<p>Source, allowed values are:</p> <ul style="list-style-type: none"> • "" default value. • "okm:folder" indicate the value is a folder uuid. <div>  When type "okm:folder" is applied will be shown a folder selector popup. </div>
AMD_TYPE01	<p>Allowed type of values are:</p> <ul style="list-style-type: none"> • "" default value. • "text". • "textarea". • "integer". • "boolean". <div>  The type will take effect on widget shown from UI. </div>
AMD_TYPE02	<p>Allowed type of values are:</p> <ul style="list-style-type: none"> • "" default value. • "text". • "textarea". • "integer". • "boolean". <div>  The type will take effect on widget shown from UI. </div>
AMD_TYPE03	<p>Allowed type of values are:</p>

- "" default value.
- "text".
- "textarea".
- "integer".
- "boolean".



The type will take effect on widget shown from UI.



In case your Automation Action can be executed from "pre" and "post" stages, you must do an insert for each one.

Example

AddKeyword class:

```
package com.openkm.automation.action;

import java.util.HashMap;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import com.openkm.automation.Action;
import com.openkm.automation.AutomationUtils;
import com.openkm.core.Config;
import com.openkm.dao.NodeBaseDAO;

public class AddKeyword implements Action {
    private static Logger log = LoggerFactory.getLogger(AddKeyword.class);

    @Override
    public void executePre(HashMap<String, Object> env, Object... params) {
    }

    @Override
    public void executePost(HashMap<String, Object> env, Object... params) {
        String keyword = AutomationUtils.getString(0, params);
        String uuid = AutomationUtils.getUuid(env);

        try {
            if (uuid != null) {
                if (Config.SYSTEM_KEYWORD_LOWERCASE && keyword != null) {
                    keyword = keyword.toLowerCase();
                }

                NodeBaseDAO.getInstance().addKeyword(uuid, keyword);
            }
        } catch (Exception e) {
            log.error(e.getMessage(), e);
        }
    }
}
```

Go to Administration > [Database query](#) and execute the sql:

```
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
```



Depending on your database the query might have minimal changes.

Oracle:

```
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_
```

PostgreSQL:

```
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_
```

The OKM_AUTO_METADATA columns values for AddKeyword Automation Action:

Column	Description
AMD_ID	Unique Id
AMD_AT	"post"
AMD_CLASS_NAME	"com.openkm.automation.action.AddKeyword"
AMD_DESC00	"Keyword"
AMD_DESC01	""
AMD_DESC02	""
AMD_GROUP	"action".
AMD_NAME	"AddKeyword".
AMD_SRC00	"".

<i>AMD_SRC01</i>	<i>"".</i>
<i>AMD_SRC02</i>	<i>"".</i>
<i>AMD_TYPE01</i>	<i>"text".</i>
<i>AMD_TYPE02</i>	<i>"".</i>
<i>AMD_TYPE03</i>	<i>"".</i>

Creating your own Automation Validation (deprecated)



Automation architecture from version 6.3.7 has been changed instead of plugin architecture. This documentation section is for OpenKM 6.3.6 and older.

Users from version 6.3.7 should use this documentation section [Creating your own Automation Validation](#).

You can create your own Automation Validation.

Conditions:

- The new Automation Validation class **must implement the "Validation" interface**.
- The new Automation Validation class **must be declared under the package "com.openkm.automation.validation"**.



The new Automation Action class must be **within the OpenKM source code**. You might be interested in [OpenKM portable dev environment](#) for it.

Validation interface:

```
package com.openkm.automation;

import java.util.HashMap;

public interface Validation {
    public static final String METHOD = "isValid";

    boolean isValid(HashMap<String, Object> env, Object... params);
}
```

Methods description

Method	Type	Description
isValid(Map<String, Object> env, Object... params)	boolean	The method executed by the Automation event to evaluate if conditions are true or false.

Understanding env variable

The env variable is present on isValid method, it is a Map of values injected by automation. These map values provide information about the node involved on event and other related information.



It takes some time, until you take control of AutomationUtils class, because they are centralized, the methods for retrieve data.

For example to retrieve the uuid of the node that caused the event.

```
String uuid = AutomationUtils.getUuid(env);
```

Understanding params variable

The env variable is present on isValid method, it is an array of Objects. This array is filled by Automation event based on the number of parameters set on Validation class.

The array can have several Objects with a distinct type, it is necessary to retrieve the parameters to the correct object type.



It takes some time, until you take control of AutomationUtils class, because they are centralized, the methods to retrieve parameters.

For example to retrieve the first parameters as String object.




```
String param00 = AutomationUtils.getString(0, params);
```





Register

Once you have created the new Automation Validation, build the OpenKM.war file and deployed into the tomcat, it must be registered into the OKM_AUTO_METADATA table;

The OKM_AUTO_METADATA columns description:

Column	Description
AMD_ID	Autoincremental unique Id
AMD_AT	Indicates when the Automation Action can be executed, the values are: <ul style="list-style-type: none"> • "pre" indicate can be execute immediately before the action who has fired it. • "post" indicate can be execute after the action who has fired it.
AMD_CLASS_NAME	Java class name.
AMD_DESC00	Description of the parameter 0.
AMD_DESC01	Description of the parameter 1.
AMD_DESC02	Description of the parameter 2.

AMD_GROUP	<p>Type of automation, allowed values are:</p> <ul style="list-style-type: none"> • "validation".
AMD_NAME	<p>Name.</p>
AMD_SRC00	<p>Source, allowed values are:</p> <ul style="list-style-type: none"> • "" default value. • "okm:folder" indicate the value is a folder uuid. <div>  When type "okm:folder" is applied will be shown a folder selector popup. </div>
AMD_SRC01	<p>Source, allowed values are:</p> <ul style="list-style-type: none"> • "" default value. • "okm:folder" indicate the value is a folder uuid. <div>  When type "okm:folder" is applied will be shown a folder selector popup. </div>
AMD_SRC02	<p>Source, allowed values are:</p> <ul style="list-style-type: none"> • "" default value. • "okm:folder" indicate the value is a folder uuid. <div>  When type "okm:folder" is applied will be shown a folder selector popup. </div>
AMD_TYPE01	<p>Allowed type of values are:</p> <ul style="list-style-type: none"> • "" default value. • "text". • "textarea". • "integer". • "boolean".

	 The type will take effect on widget shown from UI.
AMD_TYPE02	<p>Allowed type of values are:</p> <ul style="list-style-type: none"> • "" default value. • "text". • "textarea". • "integer". • "boolean".  The type will take effect on widget shown from UI.
AMD_TYPE03	<p>Allowed type of values are:</p> <ul style="list-style-type: none"> • "" default value. • "text". • "textarea". • "integer". • "boolean".  The type will take effect on widget shown from UI.
 In case your Automation Validation can be executed from "pre" and "post" stages, you must do an insert for each one.	

Example

PathContains class:

```
package com.openkm.automation.validation;

import java.util.HashMap;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import com.openkm.api.OKMRepository;
import com.openkm.automation.AutomationUtils;
```

```

import com.openkm.automation.Validation;

/**
 * Check if the current parent path contains a designed one. The only
 * parameter is a path and will test if this one is included in the
 * actual parent.
 */
public class PathContains implements Validation {
    private static Logger log = LoggerFactory.getLogger(PathContains.class);

    @Override
    public boolean isValid(HashMap<String, Object> env, Object... params) {
        try {
            String uuid = AutomationUtils.getString(0, params);
            String parentPath = AutomationUtils.getParentPath(env);
            String path = OKMRepository.getInstance().getNodePath(null, uuid);

            if (parentPath.startsWith(path)) {
                return true;
            } else {
                return false;
            }
        } catch (Exception e) {
            log.error(e.getMessage(), e);
        }

        return false;
    }
}

```

Go to Administration > [Database query](#) and execute the sql:

```
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_GROUP, AMD_TYPE)
```



Depending on your database the query might have minimal changes.

Oracle:

```
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_
```

PostgreSQL:

```
INSERT INTO OKM_AUTO_METADATA (AMD_ID, AMD_AT, AMD_CLASS_NAME, AMD_NAME, AMD_
```

The OKM_AUTO_METADATA columns values for PathContains Automation Validation:

Column	Description
AMD_ID	Unique Id
AMD_AT	"post"

AMD_CLASS_NAME	"com.openkm.automation.validation.PathContains"
AMD_DESC00	"String"
AMD_DESC01	""
AMD_DESC02	""
AMD_GROUP	"validation".
AMD_NAME	"PathContains".
AMD_SRC00	"okm:folder".
AMD_SRC01	"".
AMD_SRC02	"".
AMD_TYPE01	"text".
AMD_TYPE02	"".
AMD_TYPE03	"".

Creating your own Crontab job

Application support jobs written in BeanShell or classes into JAR file. We recommend the use JAR files instead of BeanShell.

Understanding the crontab job execution context

Execution context:

- Crontab job is executed by internal users named "system" (**the job has not a valid user session**).
- Crontab job is executed by administrator user (**the job has a valid user session**).



All API methods have variable named token.

When **user is logged** the call to the API use **token value to null**. That indicates application must use the logged session.

When **user is not logged**, the call to the API must use a **not null token**, otherwise will get an error.

Example of use:

- logged user -> **valid** API call -> `OKMFolder.getInstance().createSimple(null, "/okm:root/test");`
- not logged user -> **invalid** API call -> `OKMFolder.getInstance().createSimple(null, "/okm:root/test");`

Use the system token is a good practice when calling API methods into crontab task:

```
String systemToken = DbSessionManager.getInstance().getSystemToken();
OKMFolder.getInstance().createSimple(systemToken, "/okm:root/test")
```

Crontab job based on JAR file

Requisites:

- Must implement the class "**public static void main(String[] args)**".

```
package com.openkm;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import com.openkm.module.db.stuff.DbSessionManager;

public class CronTabTest {
    private static Logger log = LoggerFactory.getLogger(CronTabTest.class);

    public static void main(String[] args) {
        System.out.println(cronTask(DbSessionManager.getInstance().getSystemToken()));
    }
}
```

To create a JAR file follow the instructions [Creating a JAR file with Eclipse](#).

Contab job based on BeanShell

Example import files from /home/scanner folder to /okm:root/Scans



Is a good practice test script from [Scripting](#) view, before registering.



In the script you have access to these variables:

- **systemToken:** The system token for this instance.
- **crontabName:** The name of the crontab defined to run this script.
- **crontabId:** The id of the crontab defined to run this script.

```
import com.openkm.module.db.stuff.*;
import com.openkm.core.*;
import com.openkm.api.OKMDocument;
import java.io.*;

String systemToken = DbSessionManager.getInstance().getSystemsystemToken();
OKMDocument document = OKMDocument.getInstance();

public class OnlyExt implements FilenameFilter {
    String ext;

    public OnlyExt(String ext) {
        this.ext = "." + ext;
    }

    public boolean accept(File dir, String name) {
        return name.endsWith(ext);
    }
}

File scans = new File("/home/scanner");
Thread.sleep(10000); // Sleep 10 seconds in case files are still being written

try {
    for (File scan : scans.listFiles(new OnlyExt("pdf"))) {
        try {
            document.createSimple(systemToken, "/okm:root/Scans/" + scan.getName(), new File(scan.delete()));
        } catch (Exception e) {
            print("Exception: " + e);
        }
    }
} catch (Exception e) {
    print("Exception: " + e);
}
```

Creating your own database tables

The application comes with a general purpose table structure for storing its own "table values".

Related tables:

Table	Description
OKM_DB_METADATA_TYPE	Used to set table metadata definition.
OKM_DB_METADATA_VALUE	Used to store tables rows.
OKM_DB_METADATA_SEQUENCE	Used to create sequences, normally used into tables.

Create a new table definition

Table **OKM_DB_METADATA_TYPE** fields description:

Field	Description
DMT_ID	Primary key of OKM_DB_METADATA_TYPE table.
DMT_REAL_COLUMN	Set the real column name on OKM_DB_METADATA_VALUE table.
DMT_TABLE	Set the table name.
DMT_TYPE	<p>Set the column type of the table.</p> <p>Available column types:</p> <ul style="list-style-type: none">• text.• boolean.• integer.• long.• file.

Example

- Table named *country*.
- Column named *country_id* of type integer goes in *col00*.
- Column named *country_names* of type text goes in *col01*.

```
DELETE FROM OKM_DB_METADATA_TYPE WHERE DMT_TABLE='country';
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
```



Oracle and PostgreSQL databases inserts needs to set *DMT_ID* column value.

Oracle:

```
DELETE FROM OKM_DB_METADATA_TYPE WHERE DMT_TABLE='country';
INSERT INTO OKM_DB_METADATA_TYPE (DMT_ID, DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE)
INSERT INTO OKM_DB_METADATA_TYPE (DMT_ID, DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE)
```

PostgreSQL:

```
DELETE FROM OKM_DB_METADATA_TYPE WHERE DMT_TABLE='country';
INSERT INTO OKM_DB_METADATA_TYPE (DMT_ID, DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE)
INSERT INTO OKM_DB_METADATA_TYPE (DMT_ID, DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE)
```

Create a new table values

Table **OKM_DB_METADATA_VALUE** fields description:

Field	Description
DMV_ID	Primary key of OKM_DB_METADATA_VALUE table.
DMV_COL00	Set the column 0 value.
DMV_COL01	Set the column 1 value.
DMV_COL02	Set the column 2 value.
DMV_COL03	Set the column 3 value.
DMV_COL04	Set the column 4 value.

DMV_COL05	Set the column 5 value.
DMV_COL06	Set the column 6 value.
DMV_COL07	Set the column 7 value.
DMV_COL08	Set the column 8 value.
DMV_COL09	Set the column 9 value.
DMV_COL10	Set the column 10 value.
DMV_COL11	Set the column 11 value.
DMV_COL12	Set the column 12 value.
DMV_COL13	Set the column 13 value.
DMV_COL15	Set the column 14 value.
DMV_TABLE	Set the table name.

Example

- Table named country.
- Column named country_id of type integer goes in col00.
- Column named country_names of type text goes in col01.

```
DELETE FROM OKM_DB_METADATA_VALUE WHERE DMV_TABLE='country';
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('country',
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('country',
```



Oracle and PostgreSQL databases inserts needs to set DMV_ID column value.

Oracle:

```
DELETE FROM OKM_DB_METADATA_VALUE WHERE DMV_TABLE='country';
```

```
INSERT INTO OKM_DB_METADATA_VALUE (DMV_ID, DMV_TABLE, DMV_COL00, DMV_COL01)
INSERT INTO OKM_DB_METADATA_VALUE (DMV_ID, DMV_TABLE, DMV_COL00, DMV_COL01)
```

PostgreSQL:

```
DELETE FROM OKM_DB_METADATA_VALUE WHERE DMV_TABLE='country';
INSERT INTO OKM_DB_METADATA_VALUE (DMV_ID, DMV_TABLE, DMV_COL00, DMV_COL01)
INSERT INTO OKM_DB_METADATA_VALUE (DMV_ID, DMV_TABLE, DMV_COL00, DMV_COL01)
```

Create a new sequence

Table **OKM_DB_METADATA_SEQUENCE** fields description:

Field	Description
DMS_ID	Primary key of OKM_DB_METADATA_SEQUENCE table.
DMS_COLUMN	Set the column name.
DMS_TABLE	Set the table name.
DMS_TABLE	Set the next sequence value.

Example

- Table named country.
- Column country_id

```
INSERT INTO OKM_DB_METADATA_SEQUENCE (DMS_TABLE, DMS_COLUMN, DMS_VALUE) VALUES ('coun
```



Oracle and PostgreSQL databases inserts needs to set DMV_ID column value.

Oracle:

```
INSERT INTO OKM_DB_METADATA_SEQUENCE (DMS_ID, DMS_TABLE, DMS_COLUMN, DMS_VALU
```

PostgreSQL:

```
INSERT INTO OKM_DB_METADATA_SEQUENCE (DMS_ID, DMS_TABLE, DMS_COLUMN, DMS_VALU
```

Code samples



Code samples below are based on [Scripting](#).

Select a query



Queries can be used the virtual column names adding character \$ at the begining, for example \$country_name.

"DatabaseMetadataUtils" class helps on building query with method "buildQuery(table, filter, order)".

```
import com.openkm.dao.DatabaseMetadataDAO;
import com.openkm.dao.bean.DatabaseMetadataValue;
import com.openkm.util.DatabaseMetadataUtils;

for (DatabaseMetadataValue dmv : DatabaseMetadataDAO.executeValueQuery(DatabaseMetada
    System.out.println(dmv);
}

for (DatabaseMetadataValue dmv : DatabaseMetadataDAO.executeValueQuery("from Database
    System.out.println(dmv);
}
```

Update query



Method "executeValueQueryUnique" of "DatabaseMetadataDAO2 class returns only a unique record.

We must be sure the query only will retrieve a single result, otherwise will be raised an error.

```
import com.openkm.dao.DatabaseMetadataDAO;
import com.openkm.dao.bean.DatabaseMetadataValue;
import com.openkm.util.DatabaseMetadataUtils;

DatabaseMetadataValue dmv = DatabaseMetadataDAO.executeValueQueryUnique(DatabaseMetad
System.out.println(dmv);
dmv.setCol01("France");
DatabaseMetadataDAO.updateValue(dmv);
System.out.println(dmv);
```

Delete query

```
import com.openkm.dao.DatabaseMetadataDAO;
import com.openkm.dao.bean.DatabaseMetadataValue;
import com.openkm.util.DatabaseMetadataUtils;

DatabaseMetadataValue dmv =DatabaseMetadataDAO.executeValueQueryUnique(DatabaseMetada
DatabaseMetadataDAO.deleteValue(dmv.getId());
```

Get next sequence value



When the method "**getNextSequenceValue**" is executed and the sequence not exists, it will be automatically created, starting with value 1.


```
import com.openkm.dao.DatabaseMetadataDAO;  
  
Long id = new Long( DatabaseMetadataDAO.getNextSequenceValue("country", "country_id")  
System.out.println(String.format("%03d", id));
```

Use your own database table from Java

Methods

The class **DatabaseMetadataUtils** has a set of methods to select, update and delete queries:

Method	return	Description
buildQuery(String table, String filter, String order)	String	Returns a hibernate selected query.
buildUpdate(String table, String values, String filter)	String	Returns a hibernate updated query.
buildDelete(String table, String filter)	String	Returns a hibernate update query.



This method will retrieve a Hibernate query where each "\$column_name" will be replaced by it's real column name.

These queries can be executed by the method **executeHQL(String query)** at **LegacyDAO** class.

```
List<Object> results = LegacyDAO.executeHQL(query);
```



The return will be a list of DatabaseMetadataValue objects.

Sample

Register database metadata table definition



Table named "security" with tree columns:

- Column name "uuid_id" of type text.
- Column name "type" of type text.
- column name "name" of type text.

Go to **Administration** > **Tools** > [Database query](#) and execute the SQL:

```
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
```

Java classes

File *ExtendedSecurity.java*:

```
public class ExtendedSecurity extends DatabaseMetadataCommon implements IsSerializable {
    public static final String TYPE_USER = "user";
    public static final String TYPE_ROLE = "role";

    // Metadata Virtual Name mapping
    public static final String MV_TABLE_NAME = "security";
    public static final String MV_COLUMN_NAME_UUID = "uuid_id";
    public static final String MV_COLUMN_NAME_TYPE = "type";
    public static final String MV_COLUMN_NAME_NAME = "name";

    private String uuid;
    private String type;
    private String name;

    @Override
    public void loadFromMap(Map<String, String> map) {
        super.loadFromMap(map);

        if (map.containsKey(MV_COLUMN_NAME_UUID)) {
            setUuid(map.get(MV_COLUMN_NAME_UUID));
        }

        if (map.containsKey(MV_COLUMN_NAME_TYPE)) {
            setType(map.get(MV_COLUMN_NAME_TYPE));
        }

        if (map.containsKey(MV_COLUMN_NAME_NAME)) {
            setName(map.get(MV_COLUMN_NAME_NAME));
        }
    }

    @Override
    public Map<String, String> restoreToMap() {
        Map<String, String> map = super.restoreToMap();

        if (uuid != null) {
            map.put(MV_COLUMN_NAME_UUID, getUuid());
        }

        if (type != null) {
            map.put(MV_COLUMN_NAME_TYPE, getType());
        }

        if (name != null) {
            map.put(MV_COLUMN_NAME_NAME, getName());
        }

        return map;
    }

    public String getUuid() {
        return uuid;
    }

    public void setUuid(String uuid) {
        this.uuid = uuid;
    }

    public String getType() {
        return type;
    }

    public void setType(String type) {
```

```

        this.type = type;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}

```

File Example.java:

```

public class Example {
    public test() {
        // get sequence
        Double seq = new Double(DatabaseMetadataDAO.getNextSequenceValue("table_name", "c
        // create value
        final ExtendedSecurity security = new ExtendedSecurity();
        security.setUuid("some uuid");
        security.setName("some name");
        security.setType(ExtendedSecurity.TYPE_ROLE);
        security.setRealTable(ExtendedSecurity.MV_TABLE_NAME);
        Double newValue = new Double(DatabaseMetadataDAO.createValue(DatabaseMetadata

        // Execute query
        String filter = "$" + ExtendedSecurity.MV_COLUMN_NAME_UUID + "='some uuid'";
        DatabaseMetadataDAO.executeValueQuery(DatabaseMetadataUtils.buildQuery(Extend

        // Update value
        security.setName("some name changed");
        DatabaseMetadataDAO.updateValue(DatabaseMetadataUtils.getDatabaseMetadataValu

        // Delete value
        DatabaseMetadataDAO.deleteValue(DatabaseMetadataUtils.getDatabaseMetadataValu

    }
}

```

Advanced

Also joins between tables can be done. The method `executeMultiValueQuery` should be used for it:

```

List<String> tables = new ArrayList<String>();
tables.add(GWTCountry.MV_TABLE_NAME);
tables.add(GWTState.MV_TABLE_NAME);

String query = "from DatabaseMetadataValue dmv1, DatabaseMetadataValue dmv2 ";
query += "where dmv1.table='" + GWTCountry.MV_TABLE_NAME + "' ";
query += "and dmv2.table='" + GWTState.MV_TABLE_NAME + "' ";
query += "and dmv1.$" + GWTCountry.MV_COLUMN_COUNTRY_NAME + "'= 'SPAIN' ";
query += "and dmv1.$" + GWTCountry.MV_COLUMN_COUNTRY_NAME + "'=" + "dmv2.$" + GWTState

DatabaseMetadataDAO.executeMultiValueQuery(DatabaseMetadataUtils.replaceVirtual(table

```

Use your own database table from frontend (GWT)

Methods

The class **OKMDatabaseMetadataService** has the interface definition to access metadata services.

Method	return	Description
<i>executeValueQuery(String table, String filter, String order)</i>	<i>List<Map<String,String>></i>	<i>Executes a selected query.</i>
<i>updateValue(Map<String,String> map)</i>	<i>String</i>	<i>Executes an updated query.</i>
<i>createValue(Map<String,String> map)</i>	<i>String</i>	<i>Executes and creates a query.</i>
<i>executeMultiValueQuery(List<String> tables, String query)</i>	<i>List<List<Map<String, String>>></i>	<i>Executes a query that returns a results with several metadata tables.</i>
<i>getNextSequenceValue(String table, String column)</i>	<i>double</i>	<i>Gets the next value of a sequence.</i>
<i>deleteValue(Map<String, String> map)</i>	<i>void</i>	<i>Executes and deletes a query.</i>

Sample

Register database metadata table definition



Table named "security" with tree columns:

- Column name "uuid_id" of type text.
- Column name "type" of type text.
- column name "name" of type text.

Go to **Administration** > **Tools** > [Database query](#) and execute the SQL:

```
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
```

Java classes

File GWTEExtendedSecurity.java:

```
public class GWTEExtendedSecurity extends DatabaseMetadataCommon implements IsSerialized {
    public static final String TYPE_USER = "user";
    public static final String TYPE_ROLE = "role";

    // Metadata Virtual Name mapping
    public static final String MV_TABLE_NAME = "security";
    public static final String MV_COLUMN_NAME_UUID = "uuid_id";
    public static final String MV_COLUMN_NAME_TYPE = "type";
    public static final String MV_COLUMN_NAME_NAME = "name";

    private String uuid;
    private String type;
    private String name;

    @Override
    public void loadFromMap(Map<String, String> map) {
        super.loadFromMap(map);

        if (map.containsKey(MV_COLUMN_NAME_UUID)) {
            setUuid(map.get(MV_COLUMN_NAME_UUID));
        }

        if (map.containsKey(MV_COLUMN_NAME_TYPE)) {
            setType(map.get(MV_COLUMN_NAME_TYPE));
        }

        if (map.containsKey(MV_COLUMN_NAME_NAME)) {
            setName(map.get(MV_COLUMN_NAME_NAME));
        }
    }

    @Override
    public Map<String, String> restoreToMap() {
        Map<String, String> map = super.restoreToMap();

        if (uuid != null) {
            map.put(MV_COLUMN_NAME_UUID, getUuid());
        }

        if (type != null) {
            map.put(MV_COLUMN_NAME_TYPE, getType());
        }

        if (name != null) {
            map.put(MV_COLUMN_NAME_NAME, getName());
        }

        return map;
    }

    public String getUuid() {
        return uuid;
    }

    public void setUuid(String uuid) {
        this.uuid = uuid;
    }

    public String getType() {
        return type;
    }

    public void setType(String type) {
```

```

        this.type = type;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}

```

File `Example.java`:

```

public class Example {
    private final OKMDatabaseMetadataServiceAsync metadataService = (OKMDatabaseMetadataServiceAsync)
        OKMDatabaseMetadataServiceAsync.getInstance();

    public void test() {
        // get sequence
        metadataService.getNextSequenceValue("table_name", "col_name" , new AsyncCallback<Double>() {
            @Override
            public void onSuccess(Double result) {
                int value = result.intValue();
            }

            @Override
            public void onFailure(Throwable caught) {
            }
        });

        // create value
        final GWTEExtendedSecurity security = new GWTEExtendedSecurity();
        security.setUuid("some uuid");
        security.setName("some name");
        security.setType(GWTEExtendedSecurity.TYPE_ROLE);
        security.setRealTable(GWTEExtendedSecurity.MV_TABLE_NAME);
        metadataService.createValue(security.restoreToMap(), new AsyncCallback<Double>() {
            @Override
            public void onSuccess(Double result) {
                // created
            }

            @Override
            public void onFailure(Throwable caught) {
            }
        });

        // Execute query
        String filter = "$" + GWTEExtendedSecurity.MV_COLUMN_NAME_UUID + "='some uuid'";
        metadataService.executeValueQuery(GWTEExtendedSecurity.MV_TABLE_NAME, filter,
            new AsyncCallback<List<Map<String, String>>>() {
                @Override
                public void onSuccess(List<Map<String, String>> result) {
                    if (result.size() > 0) {
                        GWTEExtendedSecurity security = new GWTEExtendedSecurity();
                        security.loadFromMap(result.get(0));
                    }
                }

                @Override
                public void onFailure(Throwable caught) {
                    GeneralCommunicator.showError("executeValueQuery", caught);
                }
            });
    }
}

```

```

        // Update value
        security.setName("some name changed");
        metadataService.updateValue(security.restoreToMap(), new AsyncCallback<Object>() {
            @Override
            public void onSuccess(Object result) {
            }

            @Override
            public void onFailure(Throwable caught) {
            }
        });

        // Delete value
        metadataService.deleteValue(security.restoreToMap(), new AsyncCallback<Object>() {
            @Override
            public void onSuccess(Object result) {
            }

            @Override
            public void onFailure(Throwable caught) {
            }
        });
    }
}

```

Advanced

Also can be done joins between tables for it should be used method **executeMultiValueQuery**:

```

List<String> tables = new ArrayList<String>();
tables.add(GWTCountry.MV_TABLE_NAME);
tables.add(GWTState.MV_TABLE_NAME);

String query = "from DatabaseMetadataValue dmv1, DatabaseMetadataValue dmv2 ";
query += "where dmv1.table='" + GWTCountry.MV_TABLE_NAME + "' ";
query += "and dmv2.table='" + GWTState.MV_TABLE_NAME + "' ";
query += "and dmv1.$" + GWTCountry.MV_COLUMN_COUNTRY_NAME + "='SPAIN' ";
query += "and dmv1.$" + GWTCountry.MV_COLUMN_COUNTRY_NAME + "='" + dmv2.$" + GWTState

metadataService.executeMultiValueQuery(tables, query, new AsyncCallback<List<List<Map> > >() {
    @Override
    public void onSuccess(List<List<Map<String, String>>> result) {
        if (!result.isEmpty()) {
            List<Map<String, String>> dmvRow = result.get(0);

            if (!dmvRow.isEmpty() && dmvRow.size() == 2) {
                GWTState state = new GWTState();
                state.loadFromMap(dmvRow.get(1));
            }
        }
    }

    @Override
    public void onFailure(Throwable caught) {
    }
});

```


Creating your own login module

The class must implement *PrincipalAdapter* interface.

```
package com.openkm.principal;

import java.util.List;

public interface PrincipalAdapter {

    /**
     * Method to retrieve all users from a authentication source.
     *
     * @return A Collection with all the users.
     * @throws PrincipalAdapterException If any error occurs.
     */
    public List<String> getUsers() throws PrincipalAdapterException;

    /**
     * Method to retrieve all roles from a authentication source.
     *
     * @return A Collection with all the roles.
     * @throws PrincipalAdapterException If any error occurs.
     */
    public List<String> getRoles() throws PrincipalAdapterException;

    /**
     * Method to retrieve all users from a role.
     *
     * @return A Collection with all the users within a role.
     * @throws PrincipalAdapterException If any error occurs.
     */
    public List<String> getUsersByRole(String role) throws PrincipalAdapterException;

    /**
     * Method to retrieve all roles from a user.
     *
     * @return A Collection with all the roles of the user.
     * @throws PrincipalAdapterException If any error occurs.
     */
    public List<String> getRolesByUser(String user) throws PrincipalAdapterException;

    /**
     * Method to retrieve the mail from a user.
     *
     * @param user A user id.
     * @return The email of the user.
     * @throws PrincipalAdapterException If any error occurs.
     */
    public String getMail(String user) throws PrincipalAdapterException;

    /**
     * Method to retrieve the name from a user.
     *
     * @param user A user id.
     * @return The name of the user.
     * @throws PrincipalAdapterException If any error occurs.
     */
    public String getName(String user) throws PrincipalAdapterException;

    /**
     * Method to retrieve the user password
     *
     */
}
```

```

    * @param user A user id.
    * @return The password of the user.
    * @throws PrincipalAdapterException If any error occurs.
    */
    public String getPassword(String user) throws PrincipalAdapterException;

    /**
     * -----
     * These methods only works if using the OpenKM user database.
     * -----
     */

    /**
     * Method to create a new user
     *
     * @param user A user id.
     * @param password The password of the user.
     * @param email The user mail.
     * @param name The full user name.
     * @throws PrincipalAdapterException If any error occurs.
     */
    public void createUser(String user, String password, String email, String name, boolean active) throws PrincipalAdapterException;

    /**
     * Method to delete a user
     *
     * @param user A user id.
     * @throws PrincipalAdapterException If any error occurs.
     */
    public void deleteUser(String user) throws PrincipalAdapterException;

    /**
     * Update user information
     *
     * @param user A user id.
     * @param password The password of the user.
     * @param email The user mail.
     * @param name The full user name.
     * @throws PrincipalAdapterException If any error occurs.
     */
    public void updateUser(String user, String password, String email, String name, boolean active) throws PrincipalAdapterException;

    /**
     * Method to create a new role
     *
     * @param role A role id.
     * @throws PrincipalAdapterException If any error occurs.
     */
    public void createRole(String role, boolean active) throws PrincipalAdapterException;

    /**
     * Method to delete a role
     *
     * @param role A role id.
     * @throws PrincipalAdapterException If any error occurs.
     */
    public void deleteRole(String role) throws PrincipalAdapterException;

    /**
     * Update role information
     *
     * @param role A role id..
     * @throws PrincipalAdapterException If any error occurs.
     */
    public void updateRole(String role, boolean active) throws PrincipalAdapterException;
}

```

```

    * Method to assign a role
    *
    * @param user A user id.
    * @param role A role id.
    * @throws PrincipalAdapterException If any error occurs.
    */
    public void assignRole(String user, String role) throws PrincipalAdapterException;

    /**
    * Method to remove a role
    *
    * @param user A user id.
    * @param role A role id.
    * @throws PrincipalAdapterException If any error occurs.
    */
    public void removeRole(String user, String role) throws PrincipalAdapterException;
}

```

PrincipalAdapter sample based on properties

```

package com.openkm.principal;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Enumeration;
import java.util.List;
import java.util.Properties;
import java.util.StringTokenizer;

import org.apache.commons.lang.NotImplementedException;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import com.openkm.core.Config;

public class UsersRolesPrincipalAdapter implements PrincipalAdapter {
    private static Logger log = LoggerFactory.getLogger(UsersRolesPrincipalAdapter.class);

    @Override
    public List<String> getUsers() throws PrincipalAdapterException {
        log.debug("getUsers()");
        List<String> list = new ArrayList<String>();
        Properties prop = new Properties();

        try {
            prop.load(new FileInputStream(Config.HOME_DIR + "/server/default/conf/props/openkm"));
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }

        for (Enumeration<Object> e = prop.keys(); e.hasMoreElements();) {
            String user = (String) e.nextElement();
            if (!Config.SYSTEM_USER.equals(user)) {
                list.add(user);
            }
        }

        log.debug("getUsers: {}", list);
        return list;
    }
}

```

```

@Override
public List<String> getRoles() throws PrincipalAdapterException {
    log.debug("getRoles()");
    List<String> list = new ArrayList<String>();
    Properties prop = new Properties();

    try {
        prop.load(new FileInputStream(Config.HOME_DIR + "/server/default/conf/props/openkm
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }

    for (Enumeration<Object> e = prop.elements(); e.hasMoreElements();) {
        for (StringTokenizer st = new StringTokenizer((String) e.nextElement(), ","); st.h
            String role = st.nextToken();

            if (!Config.DEFAULT_ADMIN_ROLE.equals(role) && !list.contains(role)) {
                list.add(role);
            }
        }
    }

    log.debug("getRoles: {}", list);
    return list;
}

@Override
public List<String> getUsersByRole(String role) throws PrincipalAdapterException {
    throw new NotImplementedException("getUsersByRole");
}

@Override
public List<String> getRolesByUser(String user) throws PrincipalAdapterException {
    throw new NotImplementedException("getRolesByUser");
}

@Override
public String getMail(String user) throws PrincipalAdapterException {
    log.debug("getMail({})", user);
    Properties prop = new Properties();
    String mail = null;

    try {
        prop.load(new FileInputStream(Config.HOME_DIR + "/server/default/conf/props/openkm
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }

    mail = prop.getProperty(user);
    log.debug("getMail: {}", mail);
    return mail;
}

@Override
public String getName(String user) throws PrincipalAdapterException {
    log.debug("getName({})", user);
    Properties prop = new Properties();
    String name = null;

    try {
        prop.load(new FileInputStream(Config.HOME_DIR + "/server/default/conf/props/openkm
    } catch (FileNotFoundException e) {

```

```
e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}

name = prop.getProperty(user);
log.debug("getName: {}", name);
return name;
}

@Override
public String getPassword(String user) throws PrincipalAdapterException {
    throw new UnsupportedOperationException("Not implemented");
}

@Override
public void createUser(String user, String password, String email, String name, boolean active) throws PrincipalAdapterException {
    throw new NotImplementedException("createUser");
}

@Override
public void deleteUser(String user) throws PrincipalAdapterException {
    throw new NotImplementedException("deleteUser");
}

@Override
public void updateUser(String user, String password, String email, String name, boolean active) throws PrincipalAdapterException {
    throw new NotImplementedException("updateUser");
}

@Override
public void createRole(String role, boolean active) throws PrincipalAdapterException {
    throw new NotImplementedException("createRole");
}

@Override
public void deleteRole(String role) throws PrincipalAdapterException {
    throw new NotImplementedException("deleteRole");
}

@Override
public void updateRole(String role, boolean active) throws PrincipalAdapterException {
    throw new NotImplementedException("updateRole");
}

@Override
public void assignRole(String user, String role) throws PrincipalAdapterException {
    throw new NotImplementedException("assignRole");
}

@Override
public void removeRole(String user, String role) throws PrincipalAdapterException {
    throw new NotImplementedException("removeRole");
}
}
```

Creating your own parameters

You can take advantage of the application configuration parameters feature to store your own parameters.

To create a new configuration parameter:

- Go to **Administration > Utilities > Database query**.
- Select **JDBC option** from the list at the bottom.
- Write a query like this

```
INSERT INTO OKM_CONFIG (CFG_KEY, CFG_TYPE, CFG_VALUE) VALUES ('your.property.name', 'your.type', 'your.value');
```

- Click at **Execute button** at top right.



The **type value** - CFG_VALUE column - must be written in lower case !.

Sample

You can access it from source code to the properties.



When the property not exists will be initialized with default value (on sample code below "initial value").

```
// If the parameter not exists, by default is set to "initial value"
String value = ConfigDAO.getString("your.property.name", "initial value");
```

Creating your own password validator

You can create your own password validation. The new **password validator class must implement the `PasswordValidator` interface**:

```
public interface PasswordValidator {
    /**
     * Validate
     *
     * @param password
     * @throws ValidatorException
     */
    public void Validate(String password) throws ValidatorException;
}
```



From password validator class you can use several configuration parameters:

- `Config.PROPERTY_VALIDATOR_PASSWORD` to return value of parameter "validator.password".
- `Config.PROPERTY_VALIDATOR_PASSWORD_MIN_LENGTH` to return value of parameter "validator.password.min.length".
- `Config.PROPERTY_VALIDATOR_PASSWORD_MAX_LENGTH` to return value of parameter "validator.password.max.length".
- `Config.PROPERTY_VALIDATOR_PASSWORD_MIN_LOWERCASE` to return value of parameter "validator.password.min.lowercase".
- `Config.PROPERTY_VALIDATOR_PASSWORD_MIN_UPPERCASE` to return value of parameter "validator.password.min.uppercase".
- `Config.PROPERTY_VALIDATOR_PASSWORD_MIN_DIGITS` to return value of parameter "validator.password.min.digits".
- `Config.PROPERTY_VALIDATOR_PASSWORD_MIN_SPECIAL` to return value of parameter "validator.password.min.special".

More information at [Password validation configuration](#).

Sample

CompletePasswordValidator class:

```
/**
 * Complex password validator
 */
public class CompletePasswordValidator implements PasswordValidator {
    @SuppressWarnings("unused")
    private static Logger log = LoggerFactory.getLogger(CompletePasswordValidator.class);
}
```

```

@Override
public void Validate(String password) throws ValidatorException {
    validateLength(password);
    checkLowerCase(password);
    checkUpperCase(password);
    checkDigits(password);
    checkSpecial(password);
}

/**
 * Validate length
 */
private void validateLength(String password) throws ValidatorException {
    if (Config.VALIDATOR_PASSWORD_MIN_LENGTH > 0 &&
        password.length() < Config.VALIDATOR_PASSWORD_MIN_LENGTH) {
        throw new ValidatorException(Config.VALIDATOR_PASSWORD_ERROR_MIN_LENGTH);
    }

    if (Config.VALIDATOR_PASSWORD_MAX_LENGTH > 0 &&
        password.length() > Config.VALIDATOR_PASSWORD_MAX_LENGTH) {
        throw new ValidatorException(Config.VALIDATOR_PASSWORD_ERROR_MAX_LENGTH);
    }
}

/**
 * Validate lowercase characters
 */
private void checkLowerCase(String password) throws ValidatorException {
    int count = 0;

    if (Config.VALIDATOR_PASSWORD_MIN_LOWERCASE > 0) {
        for (int i=0; i<password.length(); i++) {
            if (Character.isLowerCase(password.charAt(i))) {
                count ++;
            }
        }

        if (Config.VALIDATOR_PASSWORD_MIN_LOWERCASE > count) {
            throw new ValidatorException(Config.VALIDATOR_PASSWORD_ERROR_MIN_LOWERCASE);
        }
    }
}

/**
 * Validate uppercase characters
 */
private void checkUpperCase(String password) throws ValidatorException {
    int count = 0;

    if (Config.VALIDATOR_PASSWORD_MIN_UPPERCASE > 0) {
        for (int i=0; i<password.length(); i++) {
            if (Character.isUpperCase(password.charAt(i))) {
                count ++;
            }
        }

        if (Config.VALIDATOR_PASSWORD_MIN_UPPERCASE > count) {
            throw new ValidatorException(Config.VALIDATOR_PASSWORD_ERROR_MIN_UPPERCASE);
        }
    }
}

/**
 * Validate digits
 */
private void checkDigits(String password) throws ValidatorException {
    int count = 0;

```



```
        if (Config.VALIDATOR_PASSWORD_MIN_DIGITS > 0) {
            for (int i=0; i<password.length(); i++) {
                if (Character.isDigit(password.charAt(i))) {
                    count ++;
                }
            }

            if (Config.VALIDATOR_PASSWORD_MIN_DIGITS > count) {
                throw new ValidatorException(Config.VALIDATOR_PASSWORD_ERROR_MIN_DIGITS);
            }
        }
    }

    /**
     * Validate special characters
     */
    private void checkSpecial(String password) throws ValidatorException {
        int count = 0;

        if (Config.VALIDATOR_PASSWORD_MIN_SPECIAL > 0) {
            for (int i=0; i<password.length(); i++) {
                if (!Character.isLetterOrDigit(password.charAt(i)) &&
                    !Character.isWhitespace(password.charAt(i))) {
                    count ++;
                }
            }

            if (Config.VALIDATOR_PASSWORD_MIN_SPECIAL > count) {
                throw new ValidatorException(Config.VALIDATOR_PASSWORD_ERROR_MIN_SPECIAL);
            }
        }
    }
}
```

Creating your own Principal Adapter

Principal adapters are used by authentication module to retrieve users and roles information. For example retrieve users and roles from some LDAP, or external database that stores all the company users and roles.

To create your own antivirus analiser must **create a new class that implements *PrincipalAdapter* interface**:

```
package com.openkm.principal;

import java.util.List;

public interface PrincipalAdapter {

    /**
     * Method to retrieve all users from a authentication source.
     *
     * @return A Collection with all the users.
     * @throws PrincipalAdapterException If any error occurs.
     */
    public List<String> getUsers() throws PrincipalAdapterException;

    /**
     * Method to retrieve all roles from a authentication source.
     *
     * @return A Collection with all the roles.
     * @throws PrincipalAdapterException If any error occurs.
     */
    public List<String> getRoles() throws PrincipalAdapterException;

    /**
     * Method to retrieve all users from a role.
     *
     * @return A Collection with all the users within a role.
     * @throws PrincipalAdapterException If any error occurs.
     */
    public List<String> getUsersByRole(String role) throws PrincipalAdapterException;

    /**
     * Method to retrieve all roles from a user.
     *
     * @return A Collection with all the roles of the user.
     * @throws PrincipalAdapterException If any error occurs.
     */
    public List<String> getRolesByUser(String user) throws PrincipalAdapterException;

    /**
     * Method to retrieve the mail from a user.
     *
     * @param user A user id.
     * @return The email of the user.
     * @throws PrincipalAdapterException If any error occurs.
     */
    public String getMail(String user) throws PrincipalAdapterException;

    /**
     * Method to retrieve the name from a user.
     *
     * @param user A user id.
     * @return The name of the user.
     * @throws PrincipalAdapterException If any error occurs.
     */
}
```

```

public String getName(String user) throws PrincipalAdapterException;

/**
 * Method to retrieve the user password
 *
 * @param user A user id.
 * @return The password of the user.
 * @throws PrincipalAdapterException If any error occurs.
 */
public String getPassword(String user) throws PrincipalAdapterException;

/*
 * -----
 * These methods only works if using the OpenKM user database.
 * -----
 */

/**
 * Method to create a new user
 *
 * @param user A user id.
 * @param password The password of the user.
 * @param email The user mail.
 * @param name The full user name.
 * @throws PrincipalAdapterException If any error occurs.
 */
public void createUser(String user, String password, String email, String name, boolean active) throws PrincipalAdapterException;

/**
 * Method to create a delete a user
 *
 * @param user A user id.
 * @throws PrincipalAdapterException If any error occurs.
 */
public void deleteUser(String user) throws PrincipalAdapterException;

/**
 * Update user information
 *
 * @param user A user id.
 * @param password The password of the user.
 * @param email The user mail.
 * @param name The full user name.
 * @throws PrincipalAdapterException If any error occurs.
 */
public void updateUser(String user, String password, String email, String name, boolean active) throws PrincipalAdapterException;

/**
 * Method to create a new role
 *
 * @param role A role id.
 * @throws PrincipalAdapterException If any error occurs.
 */
public void createRole(String role, boolean active) throws PrincipalAdapterException;

/**
 * Method to create a delete a role
 *
 * @param role A role id.
 * @throws PrincipalAdapterException If any error occurs.
 */
public void deleteRole(String role) throws PrincipalAdapterException;

/**
 * Update role information
 *
 * @param role A role id..

```

```

    * @throws PrincipalAdapterException If any error occurs.
    */
    public void updateRole(String role, boolean active) throws PrincipalAdapterException

    /**
     * Method to assign a role
     *
     * @param user A user id.
     * @param role A role id.
     * @throws PrincipalAdapterException If any error occurs.
     */
    public void assignRole(String user, String role) throws PrincipalAdapterException;

    /**
     * Method to remove a role
     *
     * @param user A user id.
     * @param role A role id.
     * @throws PrincipalAdapterException If any error occurs.
     */
    public void removeRole(String user, String role) throws PrincipalAdapterException;
}

```

The new class must be loaded into the package **com.openkm.principal** because application plugins system will try to load from there. See the sample below:




To enable the new Principal adapter go to **Administration > Configuration parameters** > find the parameter named **principal.adapter** and modify the value with **com.openkm.adapter.OwnPrincipalAdapter**

To take effect this change we need to **restart the application**.

Methods description

Method	Type	Description
getUsers()	List<String>	Return a list of users ID.
getRoles()	List<String>	Return a list of roles ID.
getUsersByRole(String role)	List<String>	Return a list of users ID filtered by a role.
getRolesByUser(String user)	List<String>	Return a list of roles ID filtered by a user.
getMail(String user)	String	Return the mail associated to a specific user.

<i>getName(String user)</i>	<i>String</i>	<i>Return the name associated to a specific user.</i>
<i>String getPassword(String user)</i>	<i>String</i>	<i>Return the password associated to a specific user.</i>
<i>createUser(String user, String password, String email, String name, boolean active)</i>	<i>void</i>	<i>Create a new user.</i>
<i>deleteUser(String user)</i>	<i>void</i>	<i>Delete a user.</i>
<i>updateUser(String user, String password, String email, String name, boolean active)</i>	<i>void</i>	<i>Update user values.</i>  <i>Password parameter can be null or empty.</i>
<i>createRole(String role, boolean active)</i>	<i>void</i>	<i>Create a new role.</i>
<i>deleteRole(String role)</i>	<i>void</i>	<i>Delete a role.</i>
<i>updateRole(String role, boolean active)</i>	<i>void</i>	<i>Update a role.</i>
<i>assignRole(String user, String role)</i>	<i>void</i>	<i>Grant user with some role.</i>
<i>removeRole(String user, String role)</i>	<i>void</i>	<i>Revoke a role to some user.</i>

Example of the Principal adapter implementation

The example retrieves users, roles and mails based on properties files.

```
package com.openkm.principal;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Enumeration;
```

```
import java.util.List;
import java.util.Properties;
import java.util.StringTokenizer;

import org.apache.commons.lang.NotImplementedException;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import com.openkm.core.Config;

public class UsersRolesPrincipalAdapter implements PrincipalAdapter {
    private static Logger log = LoggerFactory.getLogger(UsersRolesPrincipalAdapter.class);

    @Override
    public List<String> getUsers() throws PrincipalAdapterException {
        log.debug("getUsers()");
        List<String> list = new ArrayList<String>();
        Properties prop = new Properties();

        try {
            prop.load(new FileInputStream(Config.HOME_DIR + "/server/default/conf/props/openkm"));
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }

        for (Enumeration<Object> e = prop.keys(); e.hasMoreElements();) {
            String user = (String) e.nextElement();
            if (!Config.SYSTEM_USER.equals(user)) {
                list.add(user);
            }
        }

        log.debug("getUsers: {}", list);
        return list;
    }

    @Override
    public List<String> getRoles() throws PrincipalAdapterException {
        log.debug("getRoles()");
        List<String> list = new ArrayList<String>();
        Properties prop = new Properties();

        try {
            prop.load(new FileInputStream(Config.HOME_DIR + "/server/default/conf/props/openkm"));
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }

        for (Enumeration<Object> e = prop.elements(); e.hasMoreElements();) {
            for (StringTokenizer st = new StringTokenizer((String) e.nextElement(), ","); st.hasMoreTokens();) {
                String role = st.nextToken();

                if (!Config.DEFAULT_ADMIN_ROLE.equals(role) && !list.contains(role)) {
                    list.add(role);
                }
            }
        }

        log.debug("getRoles: {}", list);
        return list;
    }

    @Override
```

```
public List<String> getUsersByRole(String role) throws PrincipalAdapterException {
    throw new NotImplementedException("getUsersByRole");
}

@Override
public List<String> getRolesByUser(String user) throws PrincipalAdapterException {
    throw new NotImplementedException("getRolesByUser");
}

@Override
public String getMail(String user) throws PrincipalAdapterException {
    log.debug("getMail({})", user);
    Properties prop = new Properties();
    String mail = null;

    try {
        prop.load(new FileInputStream(Config.HOME_DIR + "/server/default/conf/props/openkm
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }

    mail = prop.getProperty(user);
    log.debug("getMail: {}", mail);
    return mail;
}

@Override
public String getName(String user) throws PrincipalAdapterException {
    log.debug("getName({})", user);
    Properties prop = new Properties();
    String name = null;

    try {
        prop.load(new FileInputStream(Config.HOME_DIR + "/server/default/conf/props/openkm
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }

    name = prop.getProperty(user);
    log.debug("getName: {}", name);
    return name;
}

@Override
public String getPassword(String user) throws PrincipalAdapterException {
    throw new UnsupportedOperationException("Not implemented");
}

@Override
public void createUser(String user, String password, String email, String name, boolean
    throw new NotImplementedException("createUser");
}

@Override
public void deleteUser(String user) throws PrincipalAdapterException {
    throw new NotImplementedException("deleteUser");
}

@Override
public void updateUser(String user, String password, String email, String name, boolean
    throw new NotImplementedException("updateUser");
}
```

```
@Override
public void createRole(String role, boolean active) throws PrincipalAdapterException {
    throw new NotImplementedException("createRole");
}

@Override
public void deleteRole(String role) throws PrincipalAdapterException {
    throw new NotImplementedException("deleteRole");
}

@Override
public void updateRole(String role, boolean active) throws PrincipalAdapterException {
    throw new NotImplementedException("updateRole");
}

@Override
public void assignRole(String user, String role) throws PrincipalAdapterException {
    throw new NotImplementedException("assignRole");
}

@Override
public void removeRole(String user, String role) throws PrincipalAdapterException {
    throw new NotImplementedException("removeRole");
}
}
```


Creating your own thesaurus

Set the configuration parameters:

Property	Value
kea.thesaurus.owl.file	vocabulary/inroute.owl
kea.thesaurus.base.url	http://www.inroutenetwork.org
kea.thesaurus.tree.root	<pre>SELECT DISTINCT UID, TEXT FROM {UID} Y {OBJECT}, {UID} rdfs:label {TEXT} ; [rdfs:subClassOf {CLAZZ}] where not bound(CLAZZ) and lang(TEXT)="en" USING NAMESPACE foaf= <http://xmlns.com/foaf/0.1/>, dcterms= <http://purl.org/dc/terms/>, rdf= <http://www.w3.org/1999/02/22-rdf-syntax-ns#>, owl= <http://www.w3.org/2002/07/owl#>, rdfs= <http://www.w3.org/2000/01/rdf-schema#>, skos= <http://www.w3.org/2004/02/skos/core#>, dc= <http://purl.org/dc/elements/1.1/></pre>
kea.thesaurus.tree.childs	<pre>SELECT DISTINCT UID, TEXT FROM {UID} rdfs:subClassOf {CLAZZ}, {UID} rdfs:label {TEXT} where xsd:string(CLAZZ) = "RDFparentID" and lang(TEXT)="en" USING NAMESPACE foaf= <http://xmlns.com/foaf/0.1/>, dcterms= <http://purl.org/dc/terms/>, rdf= <http://www.w3.org/1999/02/22-rdf-syntax-ns#>, owl= <http://www.w3.org/2002/07/owl#>, rdfs= <http://www.w3.org/2000/01/rdf-schema#>, skos= <http://www.w3.org/2004/02/skos/core#>, dc= <http://purl.org/dc/elements/1.1/></pre>

Create the inroute.owl file:

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:dcterms="http://purl.org/dc/terms/"
```

```

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:skos="http://www.w3.org/2004/02/skos/core#"
xmlns:dc="http://purl.org/dc/elements/1.1/">

<rdf:Description rdf:about="http://www.inroutenetwork.org">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Ontology"/>
  <rdfs:comment>INROUTE</rdfs:comment>
</rdf:Description>

<rdf:Description rdf:about="http://www.inroutenetwork.org#c_1">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:label xml:lang="en">Visitors trips and activities</rdfs:label>
</rdf:Description>

<rdf:Description rdf:about="http://www.inroutenetwork.org#c_2">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:label xml:lang="en">Supply and employment</rdfs:label>
</rdf:Description>

<rdf:Description rdf:about="http://www.inroutenetwork.org#c_3">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:label xml:lang="en">Tourism and territory</rdfs:label>
</rdf:Description>

<rdf:Description rdf:about="http://www.inroutenetwork.org#c_4">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:label xml:lang="en">Tourism destination management</rdfs:label>
</rdf:Description>

<rdf:Description rdf:about="http://www.inroutenetwork.org#c_5">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:label xml:lang="en">Research and tools</rdfs:label>
</rdf:Description>

<rdf:Description rdf:about="http://www.inroutenetwork.org#c_1_1">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:label xml:lang="en">Visitor</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.inroutenetwork.org#c_1"/>
</rdf:Description>

<rdf:Description rdf:about="http://www.inroutenetwork.org#c_1_2">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:label xml:lang="en">Trip</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.inroutenetwork.org#c_1"/>
</rdf:Description>

<rdf:Description rdf:about="http://www.inroutenetwork.org#c_1_3">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:label xml:lang="en">Activities</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.inroutenetwork.org#c_1"/>
</rdf:Description>

<rdf:Description rdf:about="http://www.inroutenetwork.org#c_1_4">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:label xml:lang="en">Measurement and analysis tools (A.5)</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.inroutenetwork.org#c_1"/>
</rdf:Description>

<rdf:Description rdf:about="http://www.inroutenetwork.org#c_1_5">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:label xml:lang="en">Forecast modelling (A.6)</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.inroutenetwork.org#c_1"/>
</rdf:Description>

<rdf:Description rdf:about="http://www.inroutenetwork.org#c_1_6">

```

```

<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
<rdfs:label xml:lang="en">Other accounting tools (A7)</rdfs:label>
<rdfs:subClassOf rdf:resource="http://www.inroutenetwork.org#c_1"/>
</rdf:Description>

<rdf:Description rdf:about="http://www.inroutenetwork.org#c_1_7">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
<rdfs:label xml:lang="en">Other terms</rdfs:label>
<rdfs:subClassOf rdf:resource="http://www.inroutenetwork.org#c_1"/>
</rdf:Description>

<rdf:Description rdf:about="http://www.inroutenetwork.org#c_1_8">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
<rdfs:label xml:lang="en">Forms of toruism</rdfs:label>
<rdfs:subClassOf rdf:resource="http://www.inroutenetwork.org#c_1"/>
</rdf:Description>

<rdf:Description rdf:about="http://www.inroutenetwork.org#c_2_1">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
<rdfs:label xml:lang="en">Establishments ( productive )</rdfs:label>
<rdfs:subClassOf rdf:resource="http://www.inroutenetwork.org#c_2"/>
</rdf:Description>

<rdf:Description rdf:about="http://www.inroutenetwork.org#c_2_2">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
<rdfs:label xml:lang="en">Companies</rdfs:label>
<rdfs:subClassOf rdf:resource="http://www.inroutenetwork.org#c_2"/>
</rdf:Description>


<rdf:Description rdf:about="http://www.inroutenetwork.org#c_2_3">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
<rdfs:label xml:lang="en">Tourism industries</rdfs:label>
<rdfs:subClassOf rdf:resource="http://www.inroutenetwork.org#c_2"/>
</rdf:Description>

<rdf:Description rdf:about="http://www.inroutenetwork.org#c_2_4">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
<rdfs:label xml:lang="en">Job</rdfs:label>
<rdfs:subClassOf rdf:resource="http://www.inroutenetwork.org#c_2"/>
</rdf:Description>

<rdf:Description rdf:about="http://www.inroutenetwork.org#c_2_5">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
<rdfs:label xml:lang="en">Other terms</rdfs:label>
<rdfs:subClassOf rdf:resource="http://www.inroutenetwork.org#c_2"/>
</rdf:Description>

</rdf:RDF>

```

- Create folder **\$TOMCAT_HOME/vocabulary**
- Copy the file **inroute.owl** into vocabulary folder
- Go to Administration -> Utilities -> [Thesaurus](#)
- On top right click on  **Generate button.**

Creating your own Suggestion Analyzer

To create your own Suggestion Analyzer must **create a new class that implements Suggestion interface**:

```
package com.openkm.form.suggestion;


import java.util.List;

import com.openkm.bean.form.Select;
import com.openkm.core.DatabaseException;
import com.openkm.core.PathNotFoundException;

public interface Suggestion {
    List<String> getSuggestions(String nodeId, String nodePath, Select sel) throws Pa
}
```

We suggest create the **new class into the package com.openkm.form.suggestion**.

Methods description

Method	Type	Description
getSuggestions(String nodeId, String nodePath, Select sel)	List<String>	<p>Return a list of suggested words to tag the document.</p> <div>  <p>The method need the document uuid (nodeId) and the document path (nodePath).</p> <p>Select can be used as a collection of possible values (enclosed dictionary).</p> </div>

Example of Suggestion Analyzer

```
package com.openkm.form.suggestion;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import com.openkm.bean.form.Option;
import com.openkm.bean.form.Select;
import com.openkm.core.DatabaseException;
import com.openkm.core.PathNotFoundException;
import com.openkm.dao.NodeDocumentDAO;
import com.openkm.dao.NodeDocumentVersionDAO;
import com.openkm.dao.SearchDAO;
import com.openkm.dao.bean.NodeDocument;
import com.openkm.dao.bean.NodeDocumentVersion;
import com.openkm.extractor.TextExtractorWork;
```

```
public class DocumentTermsSuggestion implements Suggestion {
    private static Logger log = LoggerFactory.getLogger(DocumentTermsSuggestion.class);

    @Override
    public List<String> getSuggestions(String nodeId, String nodePath, Select sel) throws
        DatabaseException {
        log.debug("getSuggestions({}, {}, {})", new Object[] { nodeId, nodePath, sel });
        List<String> list = new ArrayList<String>();

        try {
            if (NodeDocumentDAO.getInstance().isValid(nodeId)) {
                // Get text extraction
                if (!NodeDocumentDAO.getInstance().isTextExtracted(nodeId)) {
                    // Force text extraction
                    NodeDocumentVersion nDocVer = NodeDocumentVersionDAO.getInstance().findCurrentVersion(nodeId);
                    TextExtractorWork tew = new TextExtractorWork();
                    tew.setDocId(nodeId);
                    tew.setDocPath(nodePath);
                    tew.setDocVerId(nDocVer.getId());

                    // Execute extractor
                    NodeDocumentDAO.getInstance().textExtractorHelper(tew);
                }

                for (String term : SearchDAO.getInstance().getTerms(NodeDocument.class, nodeId)) {
                    for (Option opt : sel.getOptions()) {
                        if (term.equals(opt.getLabel().toLowerCase())) {
                            if (!list.contains(opt.getValue())) {
                                list.add(opt.getValue());
                                break;
                            }
                        }
                    }
                }
            }
        } catch (IOException e) {
            throw new SuggestionException("IOException: " + e.getMessage(), e);
        }

        return list;
    }
}
```

Creating your own Text extractor

To create your own text extractor you must **create a new class that extends AbstractTextExtractor interface**:

```
package org.apache.jackrabbit.extractor;

/**
 * Base class for text extractor implementations.
 */
public abstract class AbstractTextExtractor implements TextExtractor {

    /**
     * The supported content types by this text extractor.
     */
    private final String[] contentTypes;

    /**
     * @param contentTypes the supported content types by this text extractor.
     */
    public AbstractTextExtractor(String[] contentTypes) {
        this.contentTypes = new String[contentTypes.length];
        System.arraycopy(contentTypes, 0, this.contentTypes, 0, contentTypes.length);
    }

    /**
     * @inheritDoc
     */
    public String[] getContentTypes() {
        return contentTypes;
    }
}
```

Class example:

```
public class SomeNameExtractor extends AbstractTextExtractor {
    public Reader extractText(InputStream stream, String type, String encoding) throws IOException {
    }
}
```

Internally the `TextExtractorWork` will call the method `extractText` as is shown previously. The `AbstractTextExtractor` is the base class for text extractor implementations which implements the `TextExtractor` interface:

```
import java.io.IOException;
import java.io.InputStream;
import java.io.Reader;

/**
 * Interface for extracting text content from binary streams.
 */
public interface TextExtractor {
    String[] getContentTypes();
    Reader extractText(InputStream stream, String type, String encoding) throws IOException;
}

public class Tesseract3TextExtractor extends AbstractTextExtractor {
    public Reader extractText(InputStream stream, String type, String encoding) throws IOException {
    }
}
```

}

We suggest create the **new class into the package `com.openkm.extractor`**.

Methods description

Method	Type	Description
<code>getContentTypes()</code>	<code>String[]</code>	Returns the MIME types supported by this extractor. The returned strings must be in lower case, and the returned array must not be empty.
<code>extractText(InputStream stream, String type, String encoding)</code>	<code>String</code>	Returns a reader for the text content of the given binary document. The content type and character encoding (if available and applicable) are given as arguments.

Example of the Text extractor implementation

```
/**
 * OpenKM, Open Document Management System (http://www.openkm.com)
 * Copyright (c) 2006-2015 Paco Avila & Josep Llorca
 *
 * No bytes were intentionally harmed during the development of this application.
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License along
 * with this program; if not, write to the Free Software Foundation, Inc.,
 * 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
 */

package com.openkm.extractor;

import com.openkm.core.Config;
import com.openkm.core.DatabaseException;
import com.openkm.util.*;
import org.apache.commons.io.IOUtils;
import org.apache.jackrabbit.extractor.AbstractTextExtractor;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import java.io.*;
import java.util.HashMap;

/**
 * Text extractor for image documents.
 */
```

```

    * Use OCR from http://code.google.com/p/tesseract-ocr/
    */
public class Tesseract3TextExtractor extends AbstractTextExtractor {

    /**
     * Logger instance.
     */
    private static final Logger log = LoggerFactory.getLogger(Tesseract3TextExtractor.class);

    /**
     * Creates a new <code>TextExtractor</code> instance.
     */
    public Tesseract3TextExtractor() {
        super(new String[] { "image/tiff", "image/gif", "image/jpg", "image/jpeg", "image/p." });
    }

    // -----< TextExtractor >

    /**
     * {@inheritDoc}
     */
    public Reader extractText(InputStream stream, String type, String encoding) throws IOException {
        return extractText(Config.SYSTEM_OCR, stream, type, encoding);
    }

    /**
     * Extract text from image using Tesseract OCR
     */
    public String extractText(File input) throws IOException {
        return extractText(Config.SYSTEM_OCR, input);
    }

    /**
     * {@inheritDoc}
     */
    public Reader extractText(String ocr, InputStream stream, String type, String encoding) throws IOException {
        File tmpFileIn = null;

        try {
            // Create temp file
            tmpFileIn = FileUtils.createTempFileFromMime(type);
            FileOutputStream fos = new FileOutputStream(tmpFileIn);
            IOUtils.copy(stream, fos);
            fos.close();

            String text = extractText(ocr, tmpFileIn);
            return new StringReader(text);
        } catch (DatabaseException e) {
            log.warn("Failed to extract barcode text", e);
            throw new IOException(e.getMessage(), e);
        } finally {
            IOUtils.closeQuietly(stream);
            FileUtils.deleteQuietly(tmpFileIn);
        }
    }

    /**
     * Extract text from image using Tesseract OCR
     */
    public String extractText(String ocr, File input) throws IOException {
        if (!ocr.isEmpty()) {
            StringBuilder sb = new StringBuilder();
            String text = doOcr(ocr, input);
            sb.append(text);

            if (!Config.SYSTEM_OCR_ROTATE.isEmpty()) {
                String[] angles = Config.SYSTEM_OCR_ROTATE.split(Config.LIST_SEPARATOR);
            }
        }
    }
}

```



```

        for (String angle : angles) {
            // Rotate
            log.debug("Rotate image {} degrees", angle);
            double degree = Double.parseDouble(angle);
            ImageUtils.rotate(input, input, degree);
            text = doOcr(ocr, input);
            sb.append("\n-----\n");
            sb.append(text);
        }
    }

    return sb.toString();
} else {
    log.warn("Undefined OCR application");
    throw new IOException("Undefined OCR application");
}
}

/**
 * Performs OCR on image file
 */
public String doOcr(File tmpFileIn) throws IOException {
    return doOcr(Config.SYSTEM_OCR, tmpFileIn);
}

/**
 * Performs OCR on image file
 */
public String doOcr(String ocr, File tmpFileIn) throws IOException {
    BufferedReader stdout = null;
    FileInputStream fis = null;
    File tmpFileOut = null;
    String cmd = null;

    if (!ocr.isEmpty()) {
        try {
            // Create temp file
            tmpFileOut = File.createTempFile("okm", "");

            // Performs OCR
            HashMap<String, Object> hm = new HashMap<String, Object>();
            hm.put("fileIn", tmpFileIn.getPath());
            hm.put("fileOut", tmpFileOut.getPath());
            cmd = TemplateUtils.replace("SYSTEM_OCR", ocr, hm);
            ExecutionUtils.runCmd(cmd);

            // Read result
            fis = new FileInputStream(tmpFileOut.getPath() + ".txt");
            String text = IOUtils.toString(fis, "UTF-8");

            // Spellchecker
            if (Config.SYSTEM_OPENOFFICE_DICTIONARY.isEmpty()) {
                log.debug("TEXT: {}", text);
                return text;
            } else {
                text = DocumentUtils.spellChecker(text);
                log.debug("TEXT: {}", text);
                return text;
            }
        } catch (SecurityException e) {
            log.warn("Security exception executing command: " + cmd, e);
            throw new IOException(e.getMessage(), e);
        } catch (IOException e) {
            log.warn("IO exception executing command: " + cmd, e);
            throw new IOException(e.getMessage(), e);
        } catch (InterruptedException e) {

```

```
        log.warn("Interrupted exception executing command: " + cmd, e);
        throw new IOException(e.getMessage(), e);
    } catch (Exception e) {
        log.warn("Failed to extract OCR text", e);
        throw new IOException(e.getMessage(), e);
    } finally {
        IOUtils.closeQuietly(fis);
        IOUtils.closeQuietly(stdout);
        FileUtils.deleteQuietly(tmpFileOut);

        if (tmpFileOut != null) {
            FileUtils.deleteQuietly(new File(tmpFileOut.getPath() + ".txt"));
        }
    } else {
        log.warn("Undefined OCR application");
        throw new IOException("Undefined OCR application");
    }
}
```

Creating your own reports

OpenKM use JasperReports library for generating reports. For building reports you have a tool named iReport.



The **JasperReports Library** is the world's most popular Open Source reporting engine. It is entirely written in Java and it is able to use data coming from any kind of data source and produce pixel-perfect documents that can be viewed, printed or exported in a variety of document formats including HTML, PDF, Excel, OpenOffice and Word. More information at <http://community.jaspersoft.com/>.

iReport is the free, Open Source report designer for JasperReports and JasperReports Server. Create very sophisticated layouts containing charts, images, subreports, crosstabs and much more. Access your data through JDBC, TableModels, JavaBeans, XML, Hibernate, CSV, and custom sources. Then publish your reports as PDF, RTF, XML, XLS, CSV, HTML, XHTML, text, DOCX, or OpenOffice. More information at <http://community.jaspersoft.com/project/ireport-designer>.



In this section you will find an introduction for creating reports with JasperReports and iReports tool. Consider it as a quick introduction, not a complete guide. You must follow some tutorials, read books or papers to illustrate on it.

There are some available books and guides you should read:

- [The Definitive Guide to iReport by Giulio Toffoli](#).
- [JasperReports Ultimate Guide](#).
- [Ireport Designer Tutorials & Help](#).



With OpenKM you can use reports based on SQL or reports based on Scripting. For beginners we suggest starting with reports based on SQL.

- [Introducing to reports based on SQL](#).
- [Introducing to reports based on Java](#).
- [Introducing to reports with filtering](#).

Introducing to reports based on SQL

- **Install iReport tool.**
- **Execute iReport tool.**



OpenKM comes with an specific JasperReports Library engine what supports older version but now newer. At [Application version compatibility table](#) you have correspondence list between OpenKM and JasperReports Library engine versions.

You can set specific output version from iReport tool:

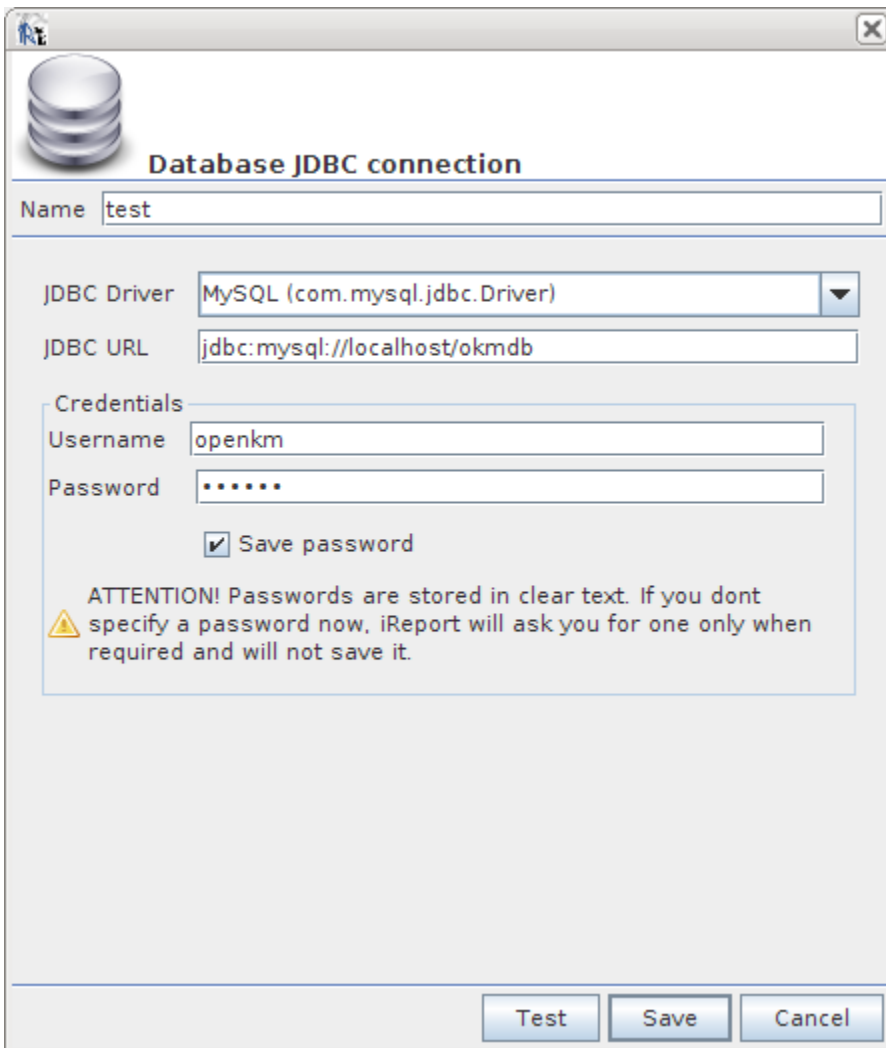
- Click on **Tools menu > options.**
- From **iReport > General tab** click on **Compatibility tab.**
- Set the **Jasper version compatibility.**
- Click on **OK button.**

Create datasource

- Click on **Report datasources button.**



- Click on **New button.**
- Choose "**Database JDBC connection**" option.
- Click on **Next button.**
- Set the **name.**
- Choose the **JDBC driver.**
- Click on **Text button** to check the connection parameters.
- Click on **Save button.**



The image shows a 'Database JDBC connection' dialog box. It has a title bar with a close button. Below the title bar is a database icon and the title 'Database JDBC connection'. The dialog contains several fields: 'Name' with the value 'test', 'JDBC Driver' with a dropdown menu showing 'MySQL (com.mysql.jdbc.Driver)', and 'JDBC URL' with the value 'jdbc:mysql://localhost/okmdb'. There is a 'Credentials' section with 'Username' set to 'openkm' and 'Password' masked with dots. A 'Save password' checkbox is checked. Below the password field is a warning message: 'ATTENTION! Passwords are stored in clear text. If you dont specify a password now, iReport will ask you for one only when required and will not save it.' At the bottom are three buttons: 'Test', 'Save', and 'Cancel'.

Database JDBC connection

Name

JDBC Driver

JDBC URL

Credentials

Username

Password

☒ Save password

ATTENTION! Passwords are stored in clear text. If you dont specify a password now, iReport will ask you for one only when required and will not save it.

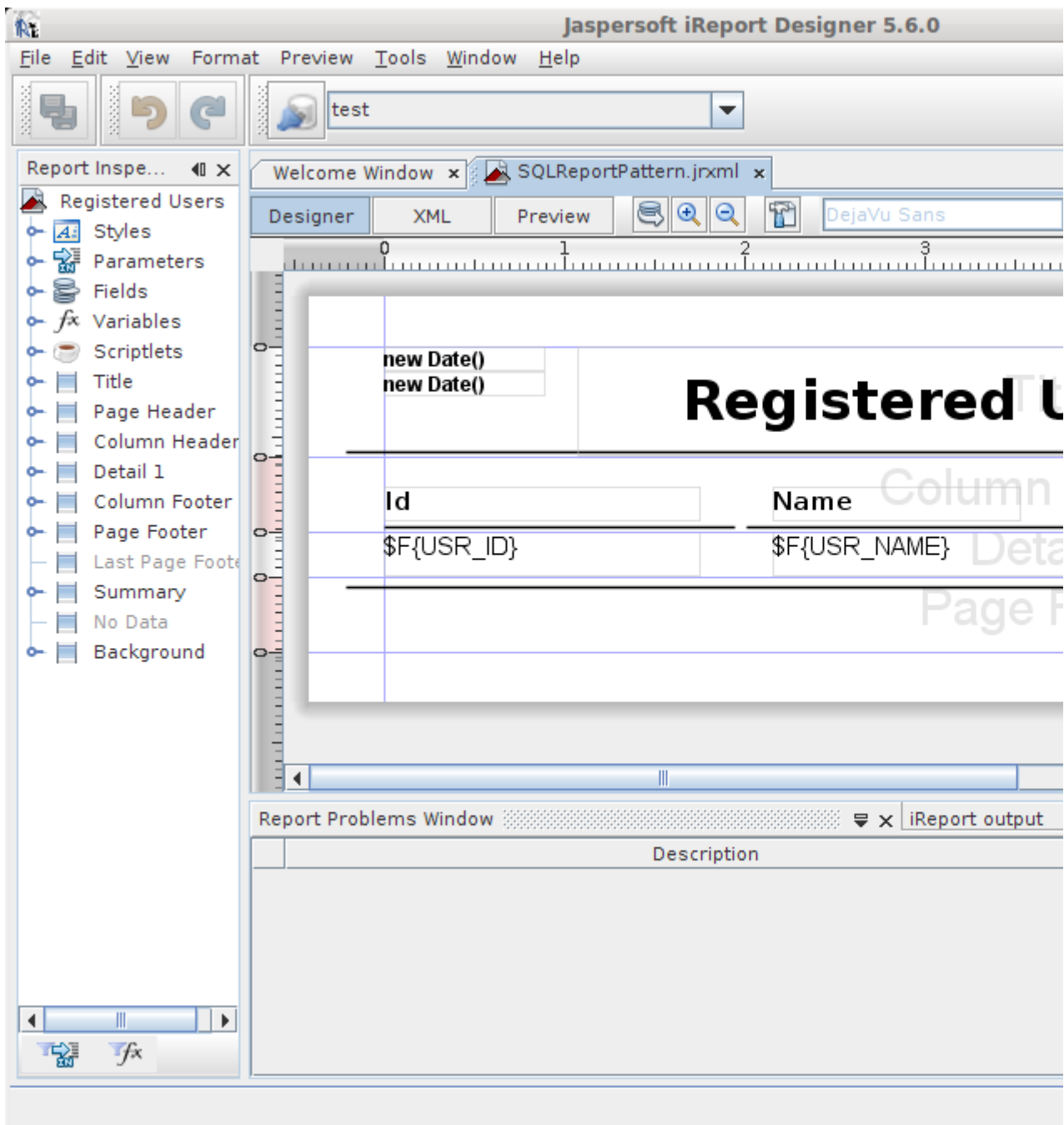
- Check as default checkbox.
- Click on *Close button*.

Testing SQL sample report

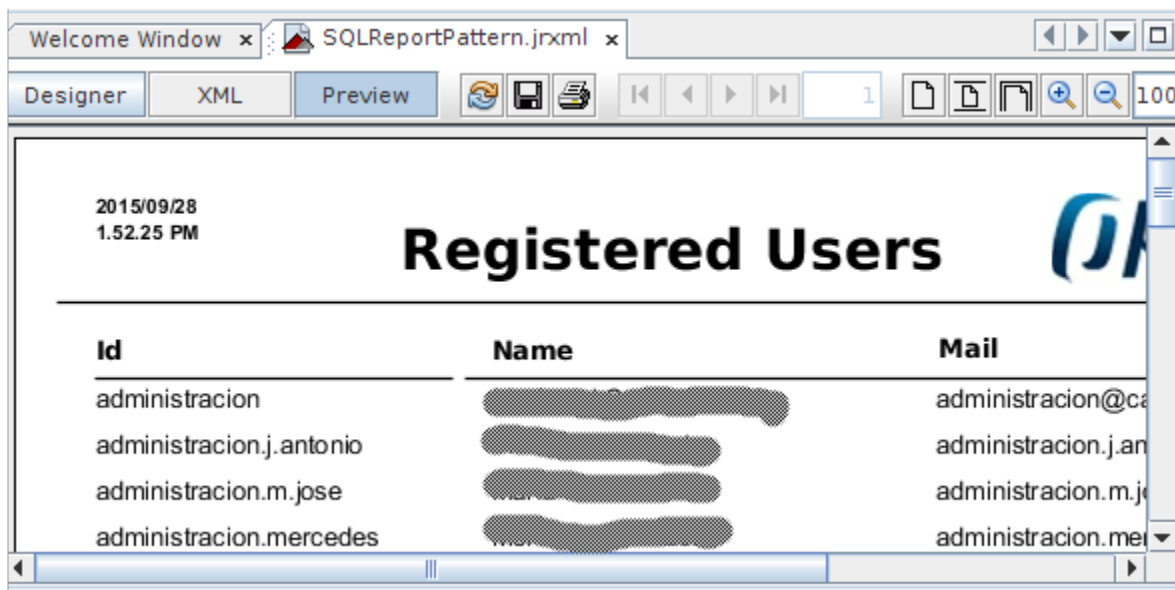


For testing the SQL Report sample you must have an OpenKM running.

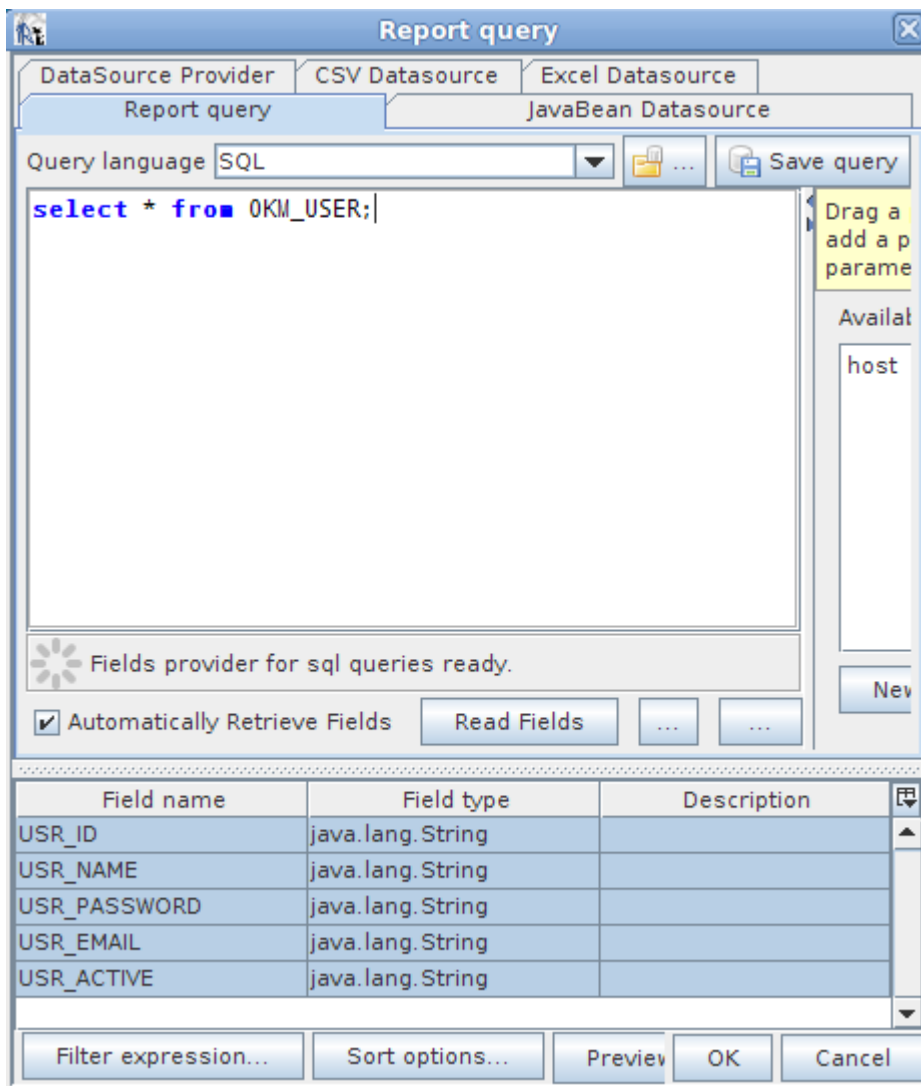
- Download the [SQLReportPattern.jrxml](#) file.
- Open the file.



- Click on **Preview** tab.
- Set the **host** value (for example <http://localhost:8080/OpenKM>).



- Click on **Designer** tab.
- Click on  **report query** button.



From this point you can add your jrxml file and register to your OpenKM at **Administration** > [Reports](#).

Closer look into JRXML file

The iReport tool is creating for you the jasper report file in XML format.



Although the iReport tool is doing all work for you in friendly user interface, is a good idea for you to take a look - at least for curiosity - about what kind of file is being created.

Query section

```
<queryString>
  <![CDATA[select * from OKM_USER;]]>
</queryString>
```


Fields section

```
<field name="USR_ID" class="java.lang.String"/>
<field name="USR_NAME" class="java.lang.String"/>
<field name="USR_PASSWORD" class="java.lang.String"/>
<field name="USR_EMAIL" class="java.lang.String"/>
<field name="USR_ACTIVE" class="java.lang.String"/>
```

Remote image section

```
<imageExpression><![CDATA[$P{host}+"/logo/report"]]></imageExpression>
```

Using fields in report

```
<textField isBlankWhenNull="false">
  <reportElement key="textField" x="329" y="0" width="206" height="18" uuid="5e3e2df8"
    <textElement>
      <font fontName="Arial"/>
    </textElement>
    <textFieldExpression><![CDATA[$F{USR_EMAIL}]]></textFieldExpression>
  </textField>
```

Introducing to reports based on Java

- Install the **iReport** tool.
- Execute **iReport** tool.



OpenKM comes with an specific JasperReports Library engine that supports older versions but now a newer one can be found. At [Application version compatibility table](#) you have correspondence list between OpenKM and JasperReports Library engine versions.

You can set specific output version from iReport tool:

- Click on **Tools menu > options**.
- From **iReport > General tab** click on **Compatibility tab**.
- Set the **Jasper version compatibility**.
- Click on **OK button**.

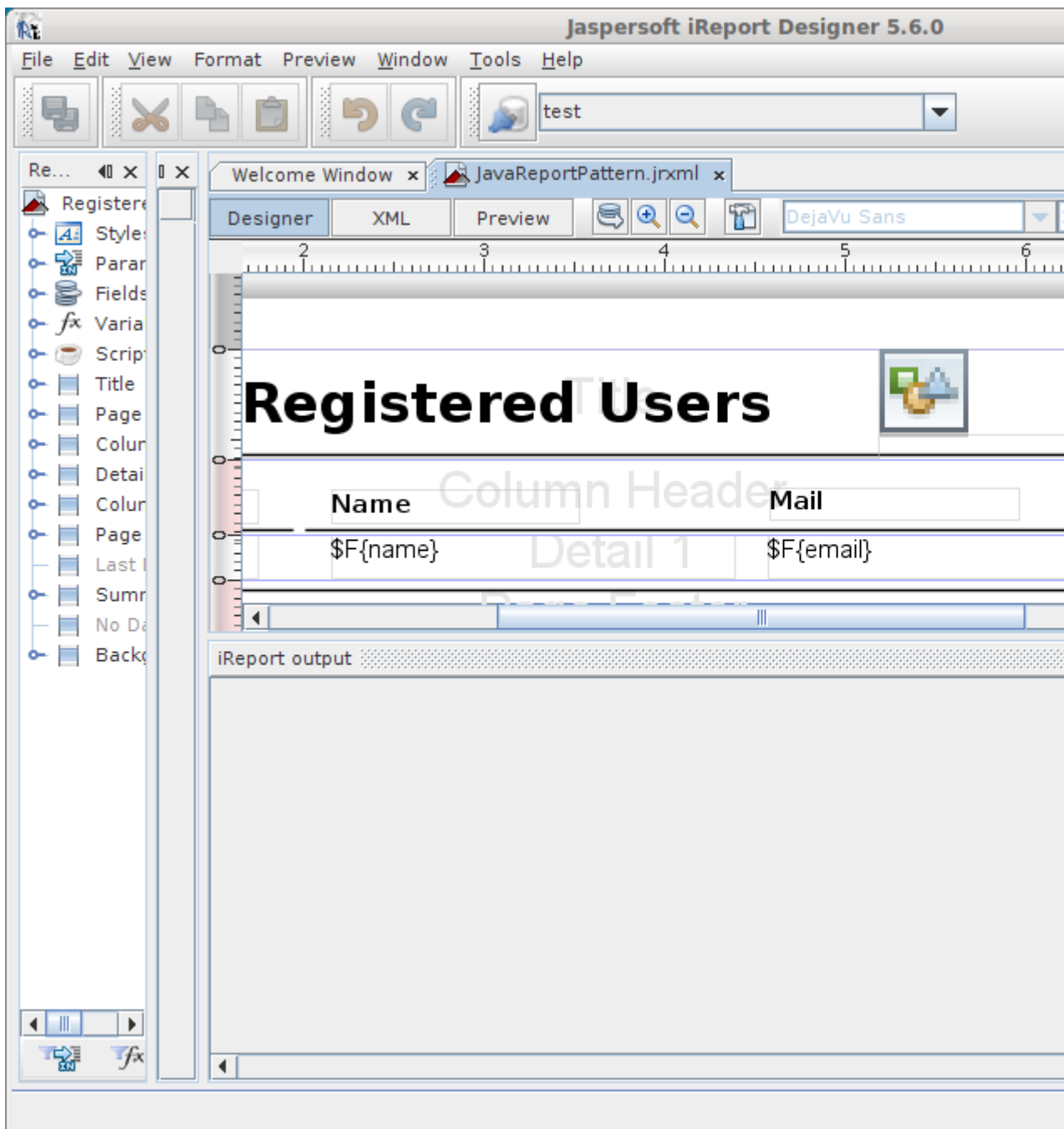
Testing Java sample report



Reports based on Java can not be previewed directly from iReport tool. Must be checked from an OpenKM.

For testing SQL Report sample you must have an OpenKM running.

- Download the [JavaReportPattern.jrxml](#) file.
- Open the file.



- Click on **Designer tab**.
- Click on  **report query button**.

Report query

DataSource Provider | CSV Datasource | Excel Datasource | **JavaBean Datasource**

Report query | JavaBean Datasource

Query language: **SQL** | Load query | Save query

```
import com.openkm.dao.*;
import com.openkm.dao.bean.*;

List al = new ArrayList();
for (User user : AuthDAO.findAllUsers(false)) {
    Map usr = new HashMap();
    usr.put("id", user.getId());
    usr.put("name", user.getName());
    usr.put("email", user.getEmail());
    usr.put("roles", user.getRoles().toString());
    al.add(usr);
}

return al;
```

Drag a p add a pa paramet

Available: host

New

Fields provider for sql queries ready.

☒ Automatically Retrieve Fields | Read Fields | Q... | Sen...

Field name	Field type	Description
name	java.lang.String	name
email	java.lang.String	email
id	java.lang.String	id

Filter expression... | Sort options... | Preview data ▼ | OK | Cancel



In this case in place of SQL query you can set a Java [Scripting](#) (see also [Scripting samples](#)).

The script must return a `List<Map>` object where each value into the map corresponds to a column value.

From this point you can add your jrxml file and register to your OpenKM at **Administration** > [Reports](#) .

Understanding the code sample

- Create an empty list.
- Iterate across all users calling the API method named `AuthDAO.findAllUsers(false)`.
- Create a Map for each User object where each value corresponds to a column.
- Add each map into the list, where each map corresponds to a row.

```
import com.openkm.dao.*;
import com.openkm.dao.bean.*;

List al = new ArrayList();
```

```
for (User user : AuthDAO.findAllUsers(false)) {  
    Map usr = new HashMap();  
    usr.put("id", user.getId());  
    usr.put("name", user.getName());  
    usr.put("email", user.getEmail());  
    usr.put("roles", user.getRoles().toString());  
    al.add(usr);  
}  
  
return al;
```

Using in report

- For each key in map create a field.
- Use the field in report

The screenshot displays the JasperSoft iReport Designer 5.6.0 interface. The 'Report Inspector' on the left lists various report components, with 'Fields' highlighted. Under 'Fields', the 'name' field is selected, indicated by a red box and an arrow pointing to the 'Name' column header in the report table. The report table has two columns: 'Name' and 'Mail'. The data row shows '\$F{name}' and '\$F{email}' respectively. The report title is 'Registered Users'.

Name	Mail
\$F{name}	\$F{email}

Introducing to reports with filtering

When you execute a report from OpenKM the view might be shown a form requiring the user to introduce values for filtering purpose. The most common case is filtering by a range of dates.



The reports with form filtering ends with extension **".rep"**. Really is a ZIP file renamed as **".rep"**.

These reports contains into two files:

- JRXML file
- File named params.xml

Download the [UsersLogin.zip](#) and take a look into (this is the original [UsersLogin.rep](#) renamed to ZIP).

Understanding filtering

For understanding form filter we will use the [UsersLogin.rep](#) report (download the [UsersLogin.zip](#)).

Params XML file

In the sample has been set two filtering fields:

Field	Type	Description
from_date	Input	Field used to filtering from date.
to_date	Input	Field used to filtering to date.

The params.xml file:

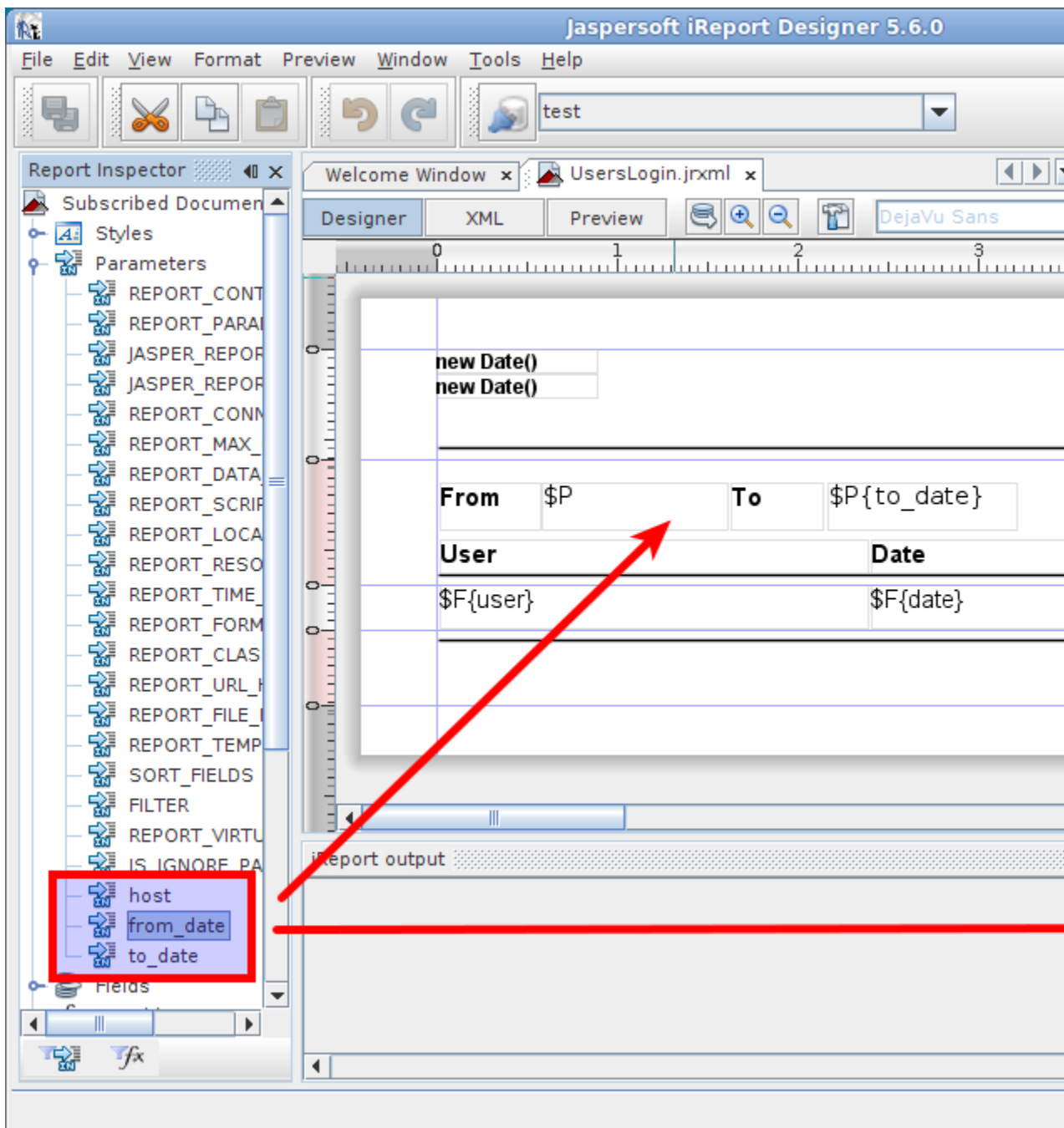
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE report-parameters PUBLIC "-//OpenKM//DTD Report Parameters 2.0//EN"
"http://www.openkm.com/dtd/report-parameters-2.0.d
<report-parameters>
  <input label="From" name="from_date" type="date">
    <validator type="req"/>
  </input>
  <input label="To" name="to_date" type="date">
    <validator type="req"/>
  </input>
</report-parameters>
```

Using fields in report

In the report are created two parameters:

Parameter	Class	Description
-----------	-------	-------------

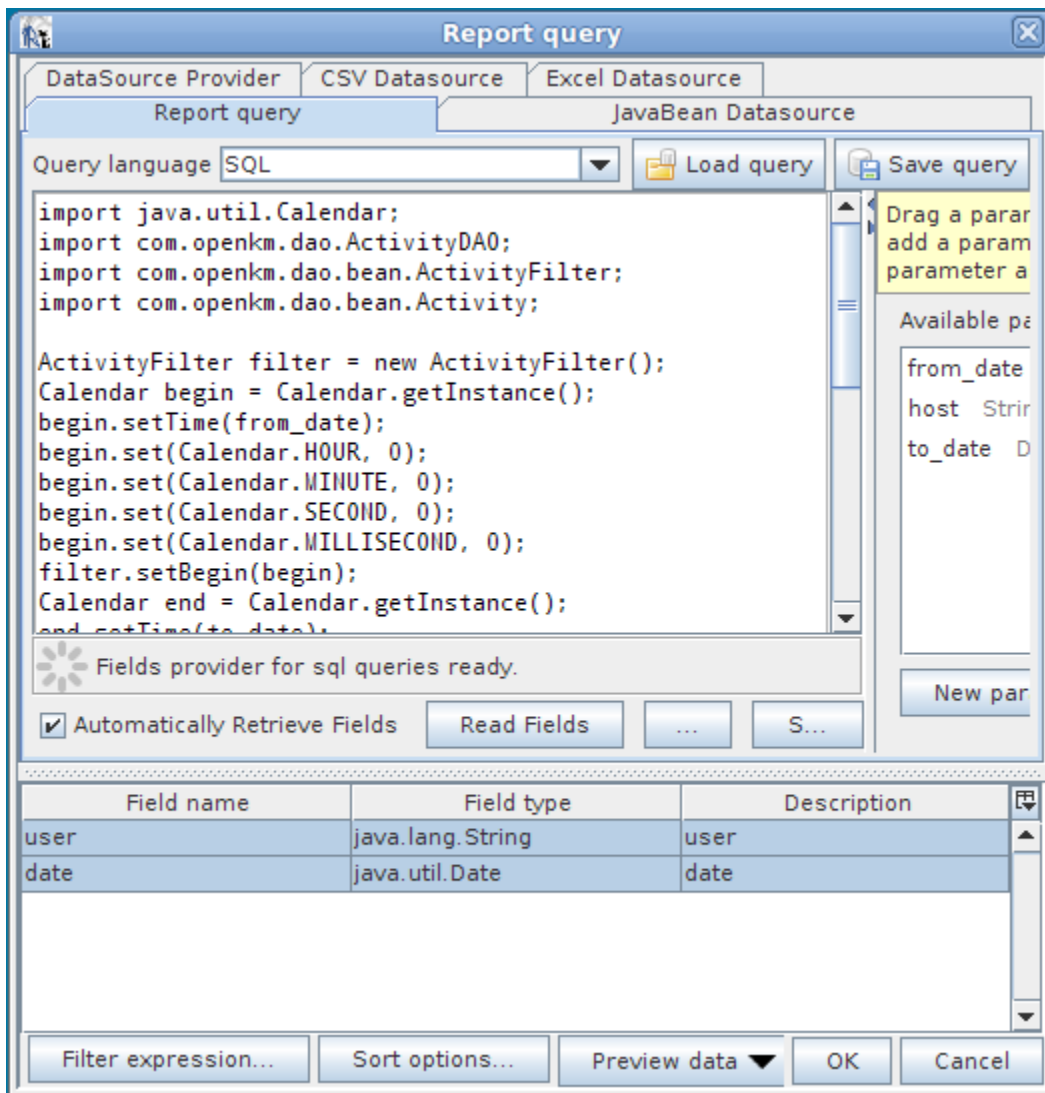
<i>from_date</i>	<i>java.util.Date</i>	<i>From date.</i>
<i>to_date</i>	<i>java.util.Date</i>	<i>To date.</i>



The parameters are also used in SQL or Java reports.

Java sample

- Click on  **report query button**.



The **from_date** and **to_date** variables are injected in code execution and can be used directly as sample below.

```
import java.util.Calendar;
import com.openkm.dao.ActivityDAO;
import com.openkm.dao.bean.ActivityFilter;
import com.openkm.dao.bean.Activity;

ActivityFilter filter = new ActivityFilter();
Calendar begin = Calendar.getInstance();
begin.setTime(from_date);
begin.set(Calendar.HOUR, 0);
begin.set(Calendar.MINUTE, 0);
begin.set(Calendar.SECOND, 0);
begin.set(Calendar.MILLISECOND, 0);
filter.setBegin(begin);
Calendar end = Calendar.getInstance();
end.setTime(to_date);
end.set(Calendar.HOUR, 0);
end.set(Calendar.MINUTE, 0);
```

```
end.set(Calendar.SECOND, 0);
end.set(Calendar.MILLISECOND, 0);
filter.setEnd(end);
filter.setAction("LOGIN");
List al = new ArrayList();

for (Iterator it = ActivityDAO.findByFilter(filter).iterator(); it.hasNext();) {
    Activity act = (Activity) it.next();
    Map actMap = new HashMap();
    actMap.put("user", act.getUser());
    actMap.put("date", act.getDate().getTime());
    al.add(actMap);
}

return al
```

SQL sample



The sample below is not based on the [UsersLogin.rep](#) report.



Click on **report query button**.

Report query

DataSource Provider CSV Datasource Excel Datasource

Report query JavaBean Datasource

Query language SQL

Load query Save query

select * from OKM_USER where USR_ID like concat('%', \$P{name}, '%');

Drag a param add a param parameter a

Available parameters

host Strir
name Str

New par

Fields provider for sql queries ready.

☒ Automatically Retrieve Fields Read Fields ... S...

Field name	Field type	Description
USR_ID	java.lang.String	
USR_ACTIVE	java.lang.String	
USR_EMAIL	java.lang.String	
USR_NAME	java.lang.String	
USR_PASSWORD	java.lang.String	
USR_PASSWORD_CHANGED	java.sql.Timestamp	

Filter expression... Sort options... Preview data OK Cancel

In the sample there is a parameter named **"name"** of class **"java.lang.String"** used for filtering users list.

```
select * from OKM_USER where USR_ID like concat('%', $P{name}, '%')
```



The engine replaces "\$P{name}" tag to "String" value. For example if parameter value is "okm" the query executed will be:

```
select * from OKM_USER where USR_ID like concat('%', 'okm', '%')
```

The contact function merges several Strings, the final result is:

```
select * from OKM_USER where USR_ID like '%okm%')
```

Params form fields

Params forms are based on formal XML definition. The DTD (Document Type Definition) defines the structure and the legal elements and attributes of an XML document.



Can see latest DTD in our website <http://www.openkm.com/dtd/report-parameters-{major}.{minor}.dtd> where {major} and {minor} values are two numbers (major release and minor release). For example the location for version 2.0 of the dtd is <http://www.openkm.com/dtd/report-parameters-2.0.dtd>



Each application version supports only a couple of DTD definitions, read [Application version compatibility table](#) to see which are supported by your application.

Basic XML skeleton

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE report-parameters PUBLIC "-//OpenKM//DTD Report Parameters 2.0//EN"
"http://www.openkm.com/dtd/report-parameters-2.0.d
<report-parameters>
</report-parameters>
```

- The **DOCTYPE** is a formal definition of some DTD. In the previous example the **report-parameters-2.0.dtd**.
- The XML begins with **<report-parameters>** tag and ends with **</report-parameters>**. Into this hierarchy are defined the metadata groups.

Basic example

Available field types:

- Checkbox
- Input
- Select
- Separator
- Suggestbox
- Text
- TextArea

More information about field type elements at [Params fields](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE report-parameters PUBLIC "-//OpenKM//DTD Report Parameters 2.0//EN"
"http://www.openkm.com/dtd/report-parameters-2.0.d
<report-parameters>
  <input label="From" name="from_date" type="date">
```

```
<validator type="req"/>
</input>
<input label="To" name="to_date" type="date">
  <validator type="req"/>
</input>
</report-parameters>
```

Using file system DTD

If the server has no access to the Internet, then the DTD file must be accessible in some way, for example can store a copy in your application server.



The XML definition must be changed to refer the new DTD location, otherwise you will get an application error while trying to process the DTD file.

Can be file system (example below) or a valid intranet URL.

```
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.0//EN"
    "file:///home/openkm/report-parameters-2.0.dtd">
```

- In the example, **/home/openkm/report-parameters-2.0.dtd** is the file system path to the file.

Params fields

These are the available field types:

Field type	Description
Checkbox	Params Checkbox field
Input	Params Input field
Select	Params Select field
Separator	Params Separator field
Suggestbox	Params Suggestbox field
Text	Params Text field
TextArea	Params TextArea field

Some field types can have children elements, validator or option elements. This hierarchy is permitted:

- *params form*
 - *Checkbox*
 - *Validator*
 - *Input*
 - *Validator*
 - *Select*
 - *Option*
 - *Validator*
 - *Separator*
 - *Suggestbox*
 - *Validator*
 - *Text*
 - *TextArea*

-
- *Validator*

Params Validator element

The goal of the Validator element is to ensure that the user provided necessary and properly formatted information needed to successfully complete an operation.

Validator can only be used in combination with other fields:

- [Params Input field](#).
- [Params Suggestbox field](#).
- [Params Checkbox field](#).
- [Params TextArea field](#).
- [Params Select field](#).

Available validators:

<i>name</i>	<i>Description</i>
req	Required value.
alpha	Alphanumeric value.
dec	Decimal value.
num	Numeric value.
email	E-mail value.
url	URL value.
maxlen	Set the maximum length allowed.
minlen	Set the minimum length required.
lt	Less than.
gt	Greater than.

min	Minimum value.
max	Maximum value.
regex	Regular expression.
Parent: <ul style="list-style-type: none"> • Params Input field. • Params Suggestbox field. • Params Checkbox field. • Params TextArea field. • Params Select field. 	

Basic example

- Value required "**req**".
- Value must be decimal "**dec**".
- The maximum lenght of the decimal value is 6 digits "**maxlen**".

```
<input label="Decimal value" name="decimal">
  <validator type="dec"/>
  <validator type="req"/>
  <validator type="maxlen" parameter="6"/>
</input>
```


Regular expression

If you want a validation which is not included in these prefefined, you can use a regular expression. For example, imagine you want to restrict an input to lowercase ASCII values:

```
<input label="Lowercase value" name="regExp">
  <validator type="regex" parameter="[a-z]+" />
</input>
```

Params Input field


in most cases is used to enter free text form, although can be used to save http links, date data or refer to application folder (as internal link to this folder).

Attribute	Description	Required
label	The text shown as label in user interface.	true
name	Unique field identifier. <div>  Two fields can't have the same name into the same task definion. Name must be unique. </div>	true
type	Type value. Allowed types: <ul style="list-style-type: none"> • text. • date. • link. • folder. 	true
width	The width of the HTML element.	false
height	The height of the HTML element.	false
Child elements: <ul style="list-style-type: none"> • Validator. For more information see Params Validator element. 		

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE report-parameters PUBLIC "-//OpenKM//DTD Report Parameters 2.1//EN"
"http://www.openkm.com/dtd/report-parameters-2.1.d
<report-parameters>
  <input label="From" name="from_date" type="date">
    <validator type="req" />
  </input>
</report-parameters>
```

Params Suggestbox field

This is an implementation of the typical AJAX suggest box component:

Attribute	Description	Required
label	The text shown as label in user interface.	true
name	Unique field identifier. <div>  Two fields can't have the same name in the same task definion. Name must be unique. </div>	true
table	Metadata table name. <div> Application can store values into metadata table, that can be used in this query. </div>	true
width	The width of the HTML element.	false
height	The height of the HTML element.	false
filterQuery	An Hibernate query to filter the data. Use {0} to place the user input.	true
valueQuery	An Hibernate query to obtain the value of a identifier. Use {0} to place the identifier.	true
filterMinLen	The minimum chars in input to begin to filter results.	true
dialogTitle	The title of the HTML dialog.	true
Child elements: <ul style="list-style-type: none"> Validator. For more information see Params Validator element. 		

Suggestbox values are defined in metadata tables.

- Application use two tables named `OKM_DB_METADATA_TYPE` AND `OKM_DB_METADATA_VALUE` that helps on storing options values used in selects. That's very useful when you got a large number of values.
- In table `OKM_DB_METADATA_TYPE` are stored metadata table definition (table name, column name and column type).
- In table `OKM_DB_METADATA_VALUE` are stored record values.

Create metadata table definition.



The **table** will be called **country**, with two columns, **country_id** and **country_name**.

The **country_id** column is **integer** type and **country_name** is **text** type.

The **country_id** is set as **column 0** with **col00** value.

The **country_name** is set as **column 1** with **col01** value.

- Go to **Administration > Utilities > Database query**.
- At bottom right select **jdbc** from list.
- **Paste the SQL** in the box

```
DELETE FROM OKM_DB_METADATA_TYPE WHERE DMT_TABLE='country';
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
```

- Click at bottom right the **Execute button**.

Insert metadata table values:

- Go to **Administration > Utilities > Database query**.
- At bottom right select **jdbc** from list.
- **Paste the SQL** in the box

```
DELETE FROM OKM_DB_METADATA_VALUE WHERE DMV_TABLE='country';
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('country',
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('country',
```

- Click at bottom right the **Execute button**.



The insert queries depending your database can not be match, for example in Oracle:

```
INSERT INTO OKM_DB_METADATA_TYPE (DMT_ID, DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE,
DMT_VIRTUAL_COLUMN) VALUES (HIBERNATE_SEQUENCE.NEXTVAL, 'country', 'col00', 'integer',
'country_id');
```

Finally the XML definition:


- In `optionsQuery` the values that begins with character `$` are identified as column names. See `$country_id` and `$country_name` below.
- Can use normal SQL clauses like **"lower"**, **"like"**, **"order by"**, etc.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE report-parameters PUBLIC "-//OpenKM//DTD Report Parameters 2.1//EN"
    "http://www.openkm.com/dtd/report-parameters-2.1.dtd"
<report-parameters>
  <suggestbox label="country" name="suggestbox" width="200px"
    table="country" dialogTitle="Choose country" filterMinLen="3"
    filterQuery="select $country_id, $country_name from DatabaseMetadataValue dmv where dmv.name=$country_name"
    valueQuery="select $country_id, $country_name from DatabaseMetadataValue dmv where dmv.name=$country_name"
  </report-parameters>
```

Params Checkbox field

Checkbox


Best used to represent boolean data:

Attribute	Description	Required
label	The text shown as label in user interface.	true
name	Unique field identifier <div>  Two fields can't have the same name in the same task definion. Name must be unique. </div>	true
width	The width of the HTML element.	false
height	The height of the HTML element.	false
Child elements: <ul style="list-style-type: none"> Validator. For more information see Params Validator element. 		

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE report-parameters PUBLIC "-//OpenKM//DTD Report Parameters 2.1//EN"
    "http://www.openkm.com/dtd/report-parameters-2.1.d
<report-parameters>
  <checkbox label="Checkbox label" name="checkbox" />
</report-parameters>
```

Params TextArea field

Print a text that can contain HTML tags.

Attribute	Description	Required
label	The text shown as label in user interface.	true
name	Unique field identifier <div>  Two fields can't have the same name in the same task definion. Name must be unique. </div>	true
width	The width of the HTML element.	false
height	The height of the HTML element.	false
Child elements: <ul style="list-style-type: none"> Validator. For more information see Params Validator element. 		

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE report-parameters PUBLIC "-//OpenKM//DTD Report Parameters 2.1//EN"
    "http://www.openkm.com/dtd/report-parameters-2.1.d
<report-parameters>
  <textarea label="textarea label" name="textarea" width="200px" />
</report-parameters>
```

Params Select field

Attribute	Description	Required
label	The text shown as label in user interface.	true
name	Unique field identifier. <div>  Two fields can't have the same name in the same task definion. Name must be unique. </div>	true
type	Type value. Allowed types: <ul style="list-style-type: none"> • simple (single choice allowed). • multiple (multiple choice allowed). 	true
table	Metadata table name. <div>  Application can store values into metadata table, that can be used in this query. </div>	*
width	The width of the HTML element.	false
height	The height of the HTML element.	false
optionsQuery	A metadata query or SQL query used to get the select options.	*
Child elements: <ul style="list-style-type: none"> • Validator. For more information see Params Validator element. 		
* There're two options to get option values: <ul style="list-style-type: none"> • Directly defined in XML. • Metadata table (table + optionsQuery). 		



One of these options must be defined, otherwise you will get an error. A combination of several of these cannot be done, only one can be chosen.

Basic select example

- Type is equal to **simple**.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE report-parameters PUBLIC "-//OpenKM//DTD Report Parameters 2.1//EN"
"http://www.openkm.com/dtd/report-parameters-2.1.d
<report-parameters>
  <select label="select label" name="select" type="simple">
    <option label="one" value="001" />
    <option label="two" value="002" />
    <option label="three" value="003" />
  </select>
</report-parameters>
```

Allow multiple choice

- Type is equal to **multiple**.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE report-parameters PUBLIC "-//OpenKM//DTD Report Parameters 2.1//EN"
"http://www.openkm.com/dtd/report-parameters-2.1.d
<report-parameters>
  <select label="select label" name="select" type="multiple" >
    <option label="one" value="001" />
    <option label="two" value="002" />
    <option label="three" value="003" />
  </select>
</report-parameters>
```

Options based on metadata table

- Application use two tables named OKM_DB_METADATA_TYPE AND OKM_DB_METADATA_VALUE that helps on storing options values used in selects. That's very useful when you got a large number of values.
- In table OKM_DB_METADATA_TYPE are stored metadata table definition (table name, column name and column type).
- In table OKM_DB_METADATA_VALUE are stored record values.

Create metadata table definition.



The **table** will be called **country**, with two columns, **country_id** and **country_name**.

The **country_id** column is **integer** type and **country_name** is **text** type.

The **country_id** is set as **column 0** with **col00** value.

*The **country_name** is set as **column 1** with **col01** value.*

- Go to **Administration > Utilities > Database query**.
- At bottom right select **JDBC** from list.
- **Paste the SQL** in the box

```
DELETE FROM OKM_DB_METADATA_TYPE WHERE DMT_TABLE='country';
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
VALUES ('country', 'col00', 'integer', 'country_id');
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
VALUES ('country', 'col01', 'integer', 'country_name');
```

- Click at bottom right the **Execute button**.

Insert metadata table values:

- Go to **Administration > Utilities > Database query**.
- At bottom right select **JDBC** from list.
- **Paste the SQL** in the box

```
DELETE FROM OKM_DB_METADATA_VALUE WHERE DMV_TABLE='country';
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('country', 'col00', 'country_id');
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('country', 'col01', 'country_name');
```

- Click at bottom right the **Execute button**.



Depending on your database the insert queries can not be matched, for example in Oracle:

```
INSERT INTO OKM_DB_METADATA_TYPE (DMT_ID, DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE,
DMT_VIRTUAL_COLUMN) VALUES (HIBERNATE_SEQUENCE.NEXTVAL, 'country', 'col00', 'integer',
'country_id');
```

More information at [Creating your own database tables](#).

Finally the XML definition:

- In optionsQuery the values that begins with character \$ are identified as column names. See **\$country_id** and **\$country_name** below.
- Can use normal sql clauses like "**order by**" etc.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE report-parameters PUBLIC "-//OpenKM//DTD Report Parameters 2.1//EN"
"http://www.openkm.com/dtd/report-parameters-2.1.dtd" [
<report-parameters>
  <select label="country" name="country" type="simple"
    table="country"
    optionsQuery="select $country_id, $country_name from DatabaseMetadataValue dmvt
  </report-parameters>
```

Based on SQL query

We can also get these option values using an SQL query. This SQL sentence should be put on the **optionsQuery** attribute, but the **table** attribute should be empty (otherwise this query is used as a metadata query). Note that the first element returned in every row is the option value and the second one is the option label.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE report-parameters PUBLIC "-//OpenKM//DTD Report Parameters 2.1//EN"
        "http://www.openkm.com/dtd/report-parameters-2.1.dtd"
<report-parameters>
  <select label="country" name="country" type="simple"
    optionsQuery="select CT_ID, CT_NAME from COUNTRY order by CT_NAME"/>
</report-parameters>
```


The table used in this sample is defined as:

```
create table COUNTRY (
  CT_ID varchar(2),
  CT_NAME varchar(32),
  primary key (CT_ID)
);

insert into COUNTRY (CT_ID, CT_NAME) values ('es', 'Spain');
insert into COUNTRY (CT_ID, CT_NAME) values ('pt', 'Portugal');
insert into COUNTRY (CT_ID, CT_NAME) values ('it', 'Italy');
```

Params Separator field


Draw a separator between fields.

Attribute	Description	Required
label	The text shown as label in user interface.	true
name	Unique field identifier <div>  Two fields can't have the same name in the same task definion. Name must be unique. </div>	true
width	The width of the HTML element.	false

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE report-parameters PUBLIC "-//OpenKM//DTD Report Parameters 2.1//EN"
    "http://www.openkm.com/dtd/report-parameters-2.1.d
<report-parameters>
  <input label="Input label" name="input1" />
  <separator label="Separator label" name="separator" />
  <input label="Input label" name="input2" />
</report-parameters>
```

Params Text field

Print a text that can contain HTML tags.

Attribute	Description	Required
label	The text shown as label in user interface.	true
name	Unique field identifier  Two fields can't have the same name in the same task definion. Name must be unique.	true
width	The width of the HTML element.	false
height	The height of the HTML element.	false

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE report-parameters PUBLIC "-//OpenKM//DTD Report Parameters 2.1//EN"
    "http://www.openkm.com/dtd/report-parameters-2.1.d
<report-parameters>
  <text label="text label" name="consulting_text" width="200px"/>
</report-parameters>
```

Creating your own Version Number Adapter

The version number adapter is used by the application to set the logic that takes control of document version number each time the document is updated.



Keep in mind changing version number adapter when application already has document version it means sometimes you must migrate older version numbers to the new format or a new implementation must take in consideration older ones.

To create your own Version number adapter you must **create a new class that implements VersionNumerationAdapter interface**:

```
package com.openkm.vernum;

import org.hibernate.Session;

import com.openkm.dao.bean.NodeDocument;
import com.openkm.dao.bean.NodeDocumentVersion;

/**
 * @author pavila
 * @see PlainVersionNumerationAdapter
 * @see BranchVersionNumerationAdapter
 * @see MajorMinorVersionNumerationAdapter
 * @see MajorMinorReleaseVersionNumerationAdapter
 */
public interface VersionNumerationAdapter {
    final String qs = "from NodeDocumentVersion ndv where ndv.parent=:parent and ndv.name=:name";

    /**
     * Obtain the initial version number to be set when creating a new document
     *
     * @return This first version number.
     */
    public String getInitialVersionNumber();

    /**
     * Calculate the next version number from a given one.
     *
     * @param session Hibernate session.
     * @param nDoc Document which will increase the revision number.
     * @param nDocVer Current document version node.
     * @return The new calculated version numbering.
     */
    public String getNextVersionNumber(Session session, NodeDocument nDoc, NodeDocumentVersion nDocVer);
}
```




To enable the new Version number adapter go to **Administration > Configuration parameters > find the parameter named `version.numeration.adapter` and modify the value with `com.openkm.vernum.OwnNumerationAdapter`**

To take effect this change need to **restating application**.

We suggest create the **new class into the package `com.openkm.vernum`**.

Methods description

Method	Type	Description
<i>getInitialVersionNumber()</i>	String	Return the first version of a new document. <div>  First version usually will be "1.0". </div>
<i>String getNextVersionNumber(Session session, NodeDocument nDoc, NodeDocumentVersion nDocVer, int increment)</i>	String	Calculate the next version number from a given one

Example of the Version number adapter implementation

```
package com.openkm.vernum;

import org.hibernate.Query;
import org.hibernate.Session;

import com.openkm.core.Config;
import com.openkm.dao.bean.NodeDocument;
import com.openkm.dao.bean.NodeDocumentVersion;

public class PlainVersionNumerationAdapter implements VersionNumerationAdapter {

    @Override
    public String getInitialVersionNumber() {
        return String.format(Config.VERSION_NUMERATION_FORMAT, 1);
    }

    @Override
    public String getNextVersionNumber(Session session, NodeDocument nDoc, NodeDocumentVersion nDocVer, int increment) {
        String versionNumber = nDocVer.getName();
        int nextVerNumber = Integer.parseInt(versionNumber);
        Query q = session.createQuery(qs);
        NodeDocumentVersion ndv = null;

        do {
            nextVerNumber++;
            q.setString("parent", nDoc.getUuid());
            q.setString("name", String.valueOf(nextVerNumber));
            ndv = (NodeDocumentVersion) q.setMaxResults(1).uniqueResult();
        } while (ndv != null);

        return String.format(Config.VERSION_NUMERATION_FORMAT, nextVerNumber);
    }
}
```

Database stuff

There are a couple of interesting configuration properties to be set at `$TOMCAT_HOME/OpenKM.cfg` file.

The parameter **hibernate.show_sql** helps developer showing at log the SQL sentences generated by Hibernate. By default this parameter is set to `false`.

```
hibernate.show_sql=true
```

The parameter `hibernate.statistics` helps developer enabling the Hibernate statistics. By default this parameter is set to `false`.

```
hibernate.statistics=true
```



Use these parameters meaningful only for development and fine tuning.

Webservices

OpenKM has API exposed via Webservices with:

- [CMIS](#)
- [RESTful](#).
- [SOAP](#).



[SOAP](#) API will no longer be supported, we've bit for [RESTful](#), that's actually more complete. We will continue maintaining SOAP for historical reasons, but in the future we will definitely deprecate it and it'll be removed from OpenKM.

We encourage move to [RESTful](#).

CMIS

Content Management Interoperability Services (CMIS) is an open standard that allows different content management systems to inter-operate over the Internet. Specifically, CMIS defines an abstraction layer for controlling diverse document management systems and repositories using web protocols.

OpenKM supports CMIS through the following modules:

- The [Apache Chemistry OpenCMIS library](#), which is a general-purpose Java library allowing developers to easily write CMIS clients and servers,
- Specific OpenKM OpenCMIS connector bundles, allowing OpenKM to be used as a CMIS server.

There is a CMIS interface available to access the content repository. As defined in the CMIS standard there are several protocols to connect to a CMIS server:

- Browser: <http://localhost:8080/OpenKM/cmis/browser>
- AtomPub: <http://localhost:8080/OpenKM/cmis/atom>

For a local OpenKM installation, you can point your browser to <http://localhost:8080/OpenKM/cmis/browser> to see the JSON response from the server. +

CMIS clients

- [CMIS Workbench](#).
- <http://code.google.com/p/cmispaces/> is an Adobe AIR client (change the `cmisUrl` parameter in the file `CMISpacesConfig.xml` to point it to OpenKM server).
- <http://code.google.com/p/cmis-explorer/> is an Adobe AIR client.



More information at:

- The official [OASIS CMIS TC](#).
- The CMIS 1.0 specification is available in [HTML](#) or [PDE](#).

RESTful

OpenKM has a complete API exposed via REST. This means you can call any of these API methods from any programming language, like Java, PHP or Python among others. This feature makes it possible to create a custom client, or integrate with third-party applications like a CRM or a CMS.



If you point your browser to <http://localhost:8080/OpenKM/services>, you can see all available webservice. At the beginning you'll see the SOAP API and at the bottom you will see a **Available RESTful services** section.

These URLs are protected by BASIC authentication so you need to provide an user and password to access them



We encourage using our SDK's for accessing OpenKM across REST API rather than you build your own client, more information:

- The [SDK for Java](#) of SDK for JAVA.
- The [SDK for .NET](#) of SDK for .NET.
- The [SDK for .NET](#) of SDK for PHP.

On almost samples you'll see parameters named "**docId**", "**fldId**" or "**nodeId**". The value of this parameter can be a valid node **UUID** or **path**.



Example of nodeId:

- Using UUID -> "**f123a950-0329-4d62-8328-0ff500fd42db**";
- Using path -> **"/okm:root/logo.png"**



Almost all samples use the command curl, more information at [curl tutorial with examples of usage](#).

Sample usage

To try these API methods you can use an HTTP Client library or any REST client which ease this process. Or simply you can use the **curl** command-line application. For example, you can list the children folders:

```
$ curl -u okmAdmin:admin -H "Accept: application/json" \
  http://localhost:8080/OpenKM/services/rest/folder/getChildren?fldId=3492d662-b58e-417c-85b6-930ad0c6c3cf
```

The result is:

```
{ "folders":
  { "folder":
    [
```

```

    { "author": "okmAdmin",
      "created": "2013-07-29T12:09:11+02:00",
      "path": "\/okm:root\/alfa",
      "permissions": 15,
      "subscribed": false,
      "uuid": "6b3e0531-96a9-4675-bb82-215b715b20ca",
      "hasChildren": false },
    { "author": "okmAdmin",
      "created": "2013-07-24T22:56:20+02:00",
      "path": "\/okm:root\/beta",
      "permissions": 15,
      "subscribed": false,
      "uuid": "41f1bace-58b4-41a9-b43e-dffc1ac9a954",
      "hasChildren": false }
  ]
}

```

In this case you can see the result in JSON format. Otherwise you may need an XML output, which can be forced using the 'Accept' header:

```

$ curl -u okmAdmin:admin -H "Accept: application/xml" \
http://localhost:8080/OpenKM/services/rest/folder/getChildren?fldId=3492d662-b58e-417c-85b6-930ad0c6c3cf

```

The result in XML is:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<folders>
  <folder>
    <author>okmAdmin</author>
    <created>2013-07-29T12:09:11+02:00</created>
    <path>/okm:root/alfa</path>
    <permissions>15</permissions>
    <subscribed>>false</subscribed>
    <uuid>6b3e0531-96a9-4675-bb82-215b715b20ca</uuid>
    <hasChildren>>false</hasChildren>
  </folder>
  <folder>
    <author>okmAdmin</author>
    <created>2013-07-24T22:56:20+02:00</created>
    <path>/okm:root/beta</path>
    <permissions>15</permissions>
    <subscribed>>false</subscribed>
    <uuid>41f1bace-58b4-41a9-b43e-dffc1ac9a954</uuid>
    <hasChildren>>false</hasChildren>
  </folder>
</folders>

```

This is a Java client for the same call:

```

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.Authenticator;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.PasswordAuthentication;
import java.net.URL;

```

```

public class JavaRestClient {
    public static void main(String[] args) throws Exception {
        try {
            String fldUuid = "3492d662-b58e-417c-85b6-930ad0c6c3cf";
            URL url = new URL("http://localhost:8080/OpenKM/services/rest/folder/getC
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setRequestMethod("GET");
            conn.setRequestProperty("Accept", "application/json");

            Authenticator.setDefault(new Authenticator() {
                protected PasswordAuthentication getPasswordAuthentication() {
                    return new PasswordAuthentication("okmAdmin", "admin".toCharArray
                }
            });

            if (conn.getResponseCode() == 200) {
                BufferedReader br = new BufferedReader(new InputStreamReader((conn.ge
                System.out.println("Output from Server .... \n");
                String output;

                while ((output = br.readLine()) != null) {
                    System.out.println(output);
                }
            } else {
                System.err.println("Failed : HTTP error code : " + conn.getResponseCo
            }

            conn.disconnect();
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

Folder samples

Create a new folder:

```

$ curl -u okmAdmin:admin -H "Accept: application/json" \
-X POST -H "Content-Type: application/json" -d '{okm:root/newfolder}' \
http://localhost:8080/OpenKM/services/rest/folder/createSimple

```

Document samples

To create a document, we need to provide the document binary data:

```

$ curl -u okmAdmin:admin -H "Accept: application/json" \
-X POST -F docPath=/okm:root/newDoc.txt -F content=@newDoc.txt \
http://localhost:8080/OpenKM/services/rest/document/createSimple

```

Or also from a HTML form:

```
<html>
<body>
  <form method="POST" enctype="multipart/form-data"
    action="http://localhost:8080/OpenKM/services/rest/document/createSimple">
    Select file: <input type="file" name="content" size="45"/><br/>
    Select path: <input type="text" name="docPath" value="/okm:root/newDoc.txt"/><br/>
    <input type="submit" value="Upload" />
  </form>
</body>
</html>
```

And now download it:

```
$ curl -u okmAdmin:admin \
  http://localhost:8080/OpenKM/services/rest/document/getContent?docId=58f79fa6-fe6e-4f68-8517-68a60898d122
```

Search samples

Basic search by content:

```
$ curl -u okmAdmin:admin -H "Accept: application/json" -X GET \
  http://localhost:8080/OpenKM/services/rest/search/findByContent?content=santo+grial
```

Basic search by keyword:

```
$ curl -u okmAdmin:admin -H "Accept: application/json" -X GET \
  http://localhost:8080/OpenKM/services/rest/search/findByKeywords?keyword=santo\&keyword=grial
```

Search with parameters:

```
$ curl -u okmAdmin:admin -H "Accept: application/json" -X GET \
  http://localhost:8080/OpenKM/services/rest/search/find?content=grial\&mimeType=application/pdf
```

Search with metadata:

```
$ curl -u okmAdmin:admin -H "Accept: application/json" -X GET \
  http://localhost:8080/OpenKM/services/rest/search/find?content=grial\&property='okp:name=alfa'
```

Security samples

Show granted users:

```
$ curl -u okmAdmin:admin -H "Accept: application/json" -X GET \
  http://localhost:8080/OpenKM/services/rest/auth/getGrantedUsers?nodeId=3492d662-b58e-417c-85b6-930ad0c6c3cf
```

Show granted roles:

```
$ curl -u okmAdmin:admin -H "Accept: application/json" -X GET \
http://localhost:8080/OpenKM/services/rest/auth/getGrantedRoles?nodeId=3492d662-b58e-417c-85b6-930ad0c6c3cf
```

Add a grant:

```
$ curl -v -u okmAdmin:admin -X PUT -H "Content-Type: application/x-www-form-urlencoded" \
-d user=john -d permissions=15 -d recursive=false -d nodeId=3492d662-b58e-417c-85b6-930ad0c6c3cf \
http://localhost:8080/OpenKM/services/rest/auth/grantUser
```

Revoke a grant:

```
$ curl -v -u okmAdmin:admin -X PUT -H "Content-Type: application/x-www-form-urlencoded" \
-d user=john -d permissions=15 -d recursive=false -d nodeId=3492d662-b58e-417c-85b6-930ad0c6c3cf \
http://localhost:8080/OpenKM/services/rest/auth/revokeUser
```

Metadata samples

The metadata group definition used in the samples:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.0//EN"
"http://www.openkm.com/dtd/property-groups-2.0.dtd">
<property-groups>
  <property-group label="Technology" name="okg:technology">
    <select label="Type" name="okp:technology.type" type="multiple">
      <option label="Alfa" value="t1"/>
      <option label="Beta" value="t2" />
      <option label="Omega" value="t3" />
    </select>
    <select label="Language" name="okp:technology.language" type="simple">
      <option label="Java" value="java"/>
      <option label="Python" value="python"/>
      <option label="PHP" value="php" />
    </select>
    <input label="Comment" name="okp:technology.comment"/>
    <textarea label="Description" name="okp:technology.description"/>
    <input label="Link" type="link" name="okp:technology.link"/>
  </property-group>
</property-groups>
```

Add a metadata group:

```
$ curl -u okmAdmin:admin -H "Accept: application/json" -X PUT \
http://localhost:8080/OpenKM/services/rest/propertyGroup/addGroup?nodeId=3492d662-b58e-417c-85b6-930ad0c6c3cf&grpName=okg:technology
```

Set metadata group values:

```
$ curl -u okmAdmin:admin -H "Accept: application/json" \
```

```
-X PUT -H "Content-Type: application/xml" \  
-d '<simplePropertiesGroup><simplePropertyGroup><name>okp:technology.comment</name><value>RESTful rulez!  
</value></simplePropertyGroup></simplePropertiesGroup>' \  
http://localhost:8080/OpenKM/services/rest/propertyGroup/setPropertiesSimple?nodeId=3492d662-b58e-417c-85b6-  
930ad0c6c3cf&grpName=okg:technology
```

Notes samples

Create a note:

```
$ curl -u okmAdmin:admin -H "Accept: application/json" \  
-X POST -H "Content-Type: application/json" -d 'Hello, world!' \  
http://localhost:8080/OpenKM/services/rest/note/add?nodeId=3492d662-b58e-417c-85b6-930ad0c6c3cf
```


Swagger



Swagger is an open source software framework backed by a large ecosystem of tools that helps developers design, build, document, and consume RESTful Web services. While most users identify Swagger by the Swagger UI tool, the Swagger toolset includes support for automated documentation, code generation, and test case generation.

How to accessing Swagger



In the sample below we use local OpenKM demo URL. You should use your own OpenKM URL (associated IP or domain).

For accessing Swagger documentation, open in your browser the URL <http://localhost:8080/OpenKM/services/> you will get some page as image below:


Available RESTful services:

Endpoint address: <http://localhost:8080/OpenKM/services/rest>

Swagger : <http://localhost:8080/OpenKM/services/rest/api-docs?url=/OpenKM/services/rest/swagger.json>

Click in the URL <http://localhost:8080/OpenKM/services/rest/api-docs?url=/OpenKM/services/rest/swagger.json> it will be shown and screen like shown below asking for authorized user credentials:

Identificación requerida


 https://demo.openkm.com solicita su nombre de usuario y contraseña. El sitio dice: "Spring Security Application"

Nombre de usuario:

Contraseña:

Cancelar Aceptar

Then will be shown the Swagger tool:

 **swagger**

/OpenKM/services/rest/swagger.json

OpenKM REST API 6.4.45

[Base URL: /OpenKM/services/rest]
</OpenKM/services/rest/swagger.json>

Copyright © OpenKM Knowledge Management System S.L.
All rights reserved. This program and the accompanying materials are made available under the terms of the license at EULA - OpenKM End User License Agreement or other commercial agreement between OpenKM ("OpenKM") and the user of this program. Portions of this product distribution require acceptance of the product distribution.

[Commercial license of API](#)

auth-service

conversion-service

document-service

Playing with Swagger

Let's do a *quick test* of the tool *checking the getRoles method from Auth-Service*.

- *Open the Auth-Service*

auth-service

GET /auth/getMail/{user}

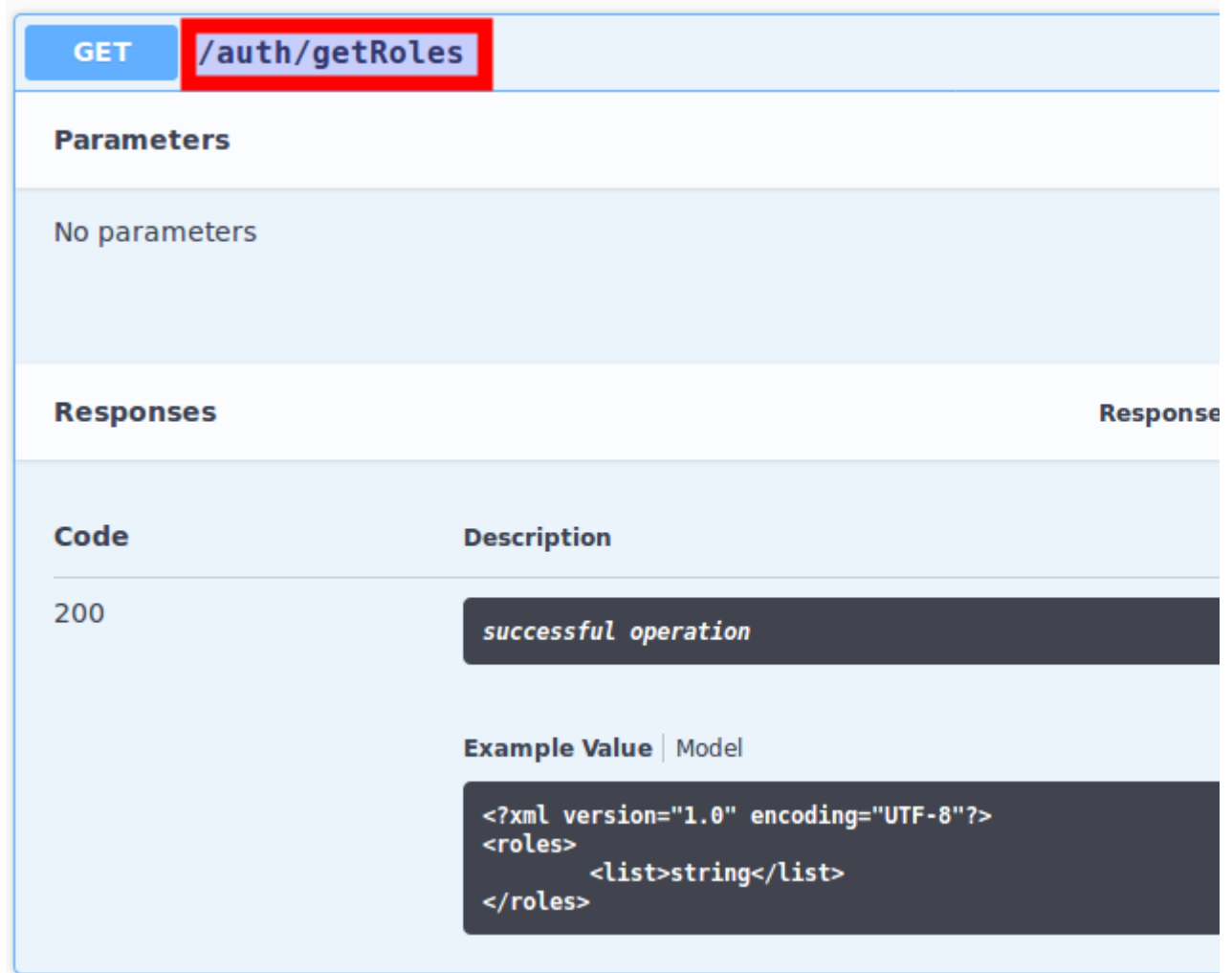
GET /auth/getUsersByRole/{role}

GET /auth/getUserProfile/{userId}

GET /auth/getProfiles/{filterByActive}

PUT /auth/changeSecurity

- *Open the `getRoles` method and click on the button "Try it out":*



GET **/auth/getRoles**

Parameters

No parameters

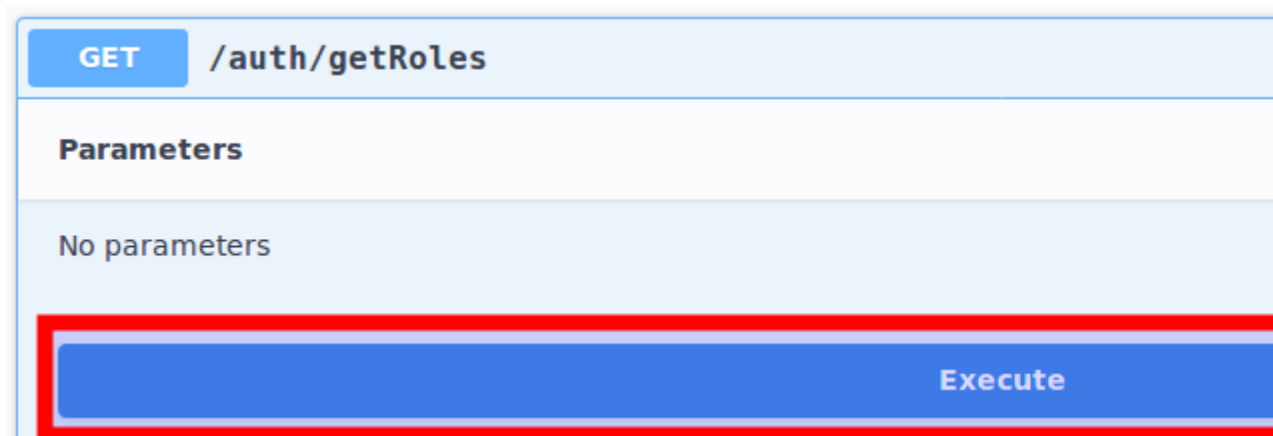
Responses

Code	Description
200	successful operation

Example Value | Model

```
<?xml version="1.0" encoding="UTF-8"?>
<roles>
  <list>string</list>
</roles>
```

- Click on the "Execute" button;



GET **/auth/getRoles**

Parameters

No parameters

Execute

- Finally will be shown the results

Responses

Response c

Curl

```
curl -X GET "http://localhost:8080/OpenKM/services/rest/auth/getRoles" -H "accept:
```

Request URL

```
http://localhost:8080/OpenKM/services/rest/auth/getRoles
```

Server response

Code	Details
200	<div><h4>Response body</h4><pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <roles> <role>ROLE_ADMIN</role> <role>ROLE_USER</role> </roles></pre></div> <div><h4>Response headers</h4><pre>content-length: 115 content-type: application/xml date: Fri, 21 Sep 2018 07:45:56 GMT</pre></div>

Responses

Code	Description
200	<pre>successful operation</pre>

SOAP

OpenKM has a limited API exposed via REST. This means you can call any of these API methods from any programming language, like Java, PHP or Python among others. This feature makes it possible to create a custom client, or integrate with third-party applications like a CRM or a CMS.



SOAP API will no longer be supported, we've bit for [RESTful](#), that's actually more complete. We will continue maintaining SOAP for historical reasons, but in the future we will definitely deprecate it and it'll be removed from OpenKM.

We encourage move to [RESTful](#).



If you point your browser to <http://localhost:8080/OpenKM/services>, you can see all available webservice. At the beginning you'll see the **Available SOAP services** section.



OpenKM SOAP WSDL follow **document / literal** style. To learn more visit [Which style of WSDL should I use?](#).

You can use MTOM. For more information read [Enabling MTOM support in JAX-WS bindings](#). The feature has been introduced from version 6.4.4.

JAVA samples



Use JAVA OpenJDK 1.8 or upper.

Create the client stuff:

```
$ wsimport -p com.openkm.ws.client -keep http://localhost:8080/OpenKM/services/OKMAuth?wsdl
```

You can use the following script to generate a complete OpenKM webservice client library:

```
#!/bin/bash

wsimport -p com.openkm.ws.client.auth -keep http://localhost:8080/OpenKM/services/OKM
wsimport -p com.openkm.ws.client.bookmark -keep http://localhost:8080/OpenKM/services
wsimport -p com.openkm.ws.client.document -keep http://localhost:8080/OpenKM/services
wsimport -p com.openkm.ws.client.folder -keep http://localhost:8080/OpenKM/services/O
wsimport -p com.openkm.ws.client.mail -keep http://localhost:8080/OpenKM/services/OKM
wsimport -p com.openkm.ws.client.note -keep http://localhost:8080/OpenKM/services/OKM
wsimport -p com.openkm.ws.client.notification -keep http://localhost:8080/OpenKM/serv
wsimport -p com.openkm.ws.client.propertyGroup -keep http://localhost:8080/OpenKM/ser
wsimport -p com.openkm.ws.client.property -keep http://localhost:8080/OpenKM/services
wsimport -p com.openkm.ws.client.repository -keep http://localhost:8080/OpenKM/servic
wsimport -p com.openkm.ws.client.search -keep http://localhost:8080/OpenKM/services/O
wsimport -p com.openkm.ws.client.workflow -keep http://localhost:8080/OpenKM/services

jar cvf okm-ws-client-6.2.jar com

rm -rf com
```

Change address dynamically from a request

```
OKMAuth okmAuth = new OKMAuth_Service(new URL(url + "/services/OKMAuth?wsdl"),
    new QName("http://ws.openkm.com", "OKMAuth")).getOKMAuthPort();
```

or

```
((BindingProvider) port).getRequestContext().put(BindingProvider.ENDPOINT_ADDRESS_PRO
```

Authentication

```
package com.openkm.ws.test;

import com.openkm.ws.client.auth.OKMAuth;
import com.openkm.ws.client.auth.OKMAuth_Service;

public class Authentication {
    public static void main(String[] args) {
        try {
            OKMAuth okmAuth = new OKMAuth_Service().getOKMAuthPort();

            // Login
            String token = okmAuth.login("okmAdmin", "admin");
            System.out.println("Token: " + token);

            // Logout
            okmAuth.logout(token);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Metadata groups

```
package com.openkm.ws.test;

import java.util.ArrayList;
import java.util.List;

import com.openkm.ws.client.auth.OKMAuth;
import com.openkm.ws.client.auth.OKMAuth_Service;
import com.openkm.ws.client.PropertyGroup;
import com.openkm.ws.client.PropertyGroup_Service;
import com.openkm.ws.client.StringPair;

public class PropertyGroups {
    public static void main(String[] args) throws Exception {
        OKMAuth okmAuth = new OKMAuth_Service().getOKMAuthPort();
        OKMPropertyGroup okmPg = new OKMPropertyGroup_Service().getOKMPropertyGroupPo

        // login
        String token = okmAuth.login("okmAdmin", "admin");

        // property groups
        List<StringPair> spList = new ArrayList<StringPair>();
        StringPair sp = new StringPair();
        sp.setKey("okp:technology.comment");
```

```

        sp.setValue("Other comment from PHP");
        spList.add(sp);
        okmPg.setPropertiesSimple(token, "/okm:root/test/hosts.txt", "okg:technology"

        // logout
        okmAuth.logout(token);
    }
}

```

.NET

Create a single webservice



.Net framework comes with "**wsdl.exe**" utility (by default it's not in your environment path).

```
$ wsdl.exe http://localhost:8080/OpenKM/services/OKMAuth?wsdl /out:AuthService.cs
```

Create all webservices in one library package

Create file called `openkm.wsdl` with contents:

```

<wsdlParameters xmlns="http://microsoft.com/webReference/">
  <nologo>true</nologo>
  <parsableerrors>true</parsableerrors>
  <sharetypes>true</sharetypes>
  <documents>
    <document>http://localhost:8080/OpenKM/services/OKMAuth?wsdl</document>
    <document>http://localhost:8080/OpenKM/services/OKMBookmark?wsdl</document>
    <document>http://localhost:8080/OpenKM/services/OKMDocument?wsdl</document>
    <document>http://localhost:8080/OpenKM/services/OKMFolder?wsdl</document>
    <document>http://localhost:8080/OpenKM/services/OKMMail?wsdl</document>
    <document>http://localhost:8080/OpenKM/services/OKMNote?wsdl</document>
    <document>http://localhost:8080/OpenKM/services/OKMNotification?wsdl</document>
    <document>http://localhost:8080/OpenKM/services/OKMPropertyGroup?wsdl</document>
    <document>http://localhost:8080/OpenKM/services/OKMProperty?wsdl</document>
    <document>http://localhost:8080/OpenKM/services/OKMRepository?wsdl</document>
    <document>http://localhost:8080/OpenKM/services/OKMSearch?wsdl</document>
    <document>http://localhost:8080/OpenKM/services/OKMWorkflow?wsdl</document>
    <document>http://localhost:8080/OpenKM/services/OKMDashboard?wsdl</document>
  </documents>
  <webReferenceOptions>
    <verbose>false</verbose>
    <codeGenerationOptions>properties newAsync enableDataBinding</codeGenerationOptions>
    <style>client</style>
  </webReferenceOptions>
</wsdlParameters>

```

Execute the command:

```
$ wsdl.exe /par:openkm.wsdl /out:Webservice.cs /n:OpenKM.ws
```



If you need a class for VB.net you have to change the command to:

```
$ wsdl.exe /l:VB /par:openkm.wsdl /out:Webservice.vb /n:OpenKM.ws
```


Will be create a file named *Webservice.cs* with namespace *OpenKM.ws* into.



For more information about *wsdl.exe* options refer to [Web Services Description Language Tool \(Wsd.exe\)](#)

Suggested changes in generated classes

Pay attention in public *AuthService* class (like others), it'll be generated some *AuthService()* constructor that you should like to change to take advantage of host variable:

```
public AuthService() {
    this.Url = http://127.0.0.1:8080/OpenKM/services/OKMAuth;
}
```

change to (or create another method with this parameter):

```
public AuthService(String host) {
    this.Url = host + "/OKMAuth";
}
```

PHP

Remote method info

```
<?php
// Register WSDL
$OKMAuth = new SoapClient('http://localhost:8080/OpenKM/services/OKMAuth?wsdl');

// Disable WSDL cache
ini_set('soap.wsdl_cache_enabled', 0);
ini_set('soap.wsdl_cache_ttl', 0);
ini_set('soap.wsdl_cache', 0);

echo "<br>**** FUNCTIONS ****<br>";
foreach ($OKMAuth->__getFunctions() as $function) {
    echo $function."<br>";
}

echo "<br>**** TYPES ****<br>";
foreach ($OKMAuth->__getTypes() as $types) {
    echo $types."<br>";
}
?>
```

Authentication

```
<?php
// Register WSDL
$OKMAuth = new SoapClient('http://localhost:8080/OpenKM/services/OKMAuth?wsdl');

// Login
$loginResp = $OKMAuth->login(array('user' => 'okmAdmin', 'password' => 'admin'));
$token = $loginResp->return;
echo "Token: ".$token;
```

```
// Logout
$OKMAuth->logout(array('token' => $token));
?>
```

List folders and documents

```
<?php
function printFolder($folder) {
    echo "[FOLDER] Path: ".$folder->path.", Author: ".$folder->author."<br>";
}

function printDocument($document) {
    echo "[DOCUMENT] Path: ".$document->path.", Author: ".$document->author.", Size:
}

// Register WSDL
$OKMAuth = new SoapClient('http://localhost:8080/OpenKM/services/OKMAuth?wsdl');
$OKMDocument = new SoapClient('http://localhost:8080/OpenKM/services/OKMDocument?wsdl');
$OKMFolder = new SoapClient('http://localhost:8080/OpenKM/services/OKMFolder?wsdl');
$path = '/okm:root';

// Login
$loginResp = $OKMAuth->login(array('user' => 'okmAdmin', 'password' => 'admin'));
$token = $loginResp->return;
echo "Token: ".$token."<br>";
echo "Path: ".$path."<br>";

// List folders
$getChildrenResp = $OKMFolder->getChildren(array('token' => $token, 'fldId' => $path));
if (isset($getChildrenResp->return)) {
    $folderArray = $getChildrenResp->return;

    if ($folderArray) {
        if (is_array($folderArray)) {
            foreach ($folderArray as $folder) {
                printFolder($folder);
            }
        } else {
            printFolder($folderArray);
        }
    }
}

// List documents
$getChildrenResp = $OKMDocument->getChildren(array('token' => $token, 'dstId' => $path));
if (isset($getChildrenResp->return)) {
    $documentArray = $getChildrenResp->return;

    if ($documentArray) {
        if (is_array($documentArray)) {
            foreach ($documentArray as $document) {
                printDocument($document);
            }
        } else {
            printDocument($documentArray);
        }
    }
}

// Logout
$OKMAuth->logout(array('token' => $token));
?>
```

Create document

```
<?php
// Register WSDL
$OKMAuth = new SoapClient('http://localhost:8080/OpenKM/services/OKMAuth?wsdl');
$OKMDocument = new SoapClient('http://localhost:8080/OpenKM/services/OKMDocument?wsdl');
$file = '/etc/hosts';

// Login
$loginResp = $OKMAuth->login(array('user' => 'okmAdmin', 'password' => 'admin'));
$token = $loginResp->return;
echo "Token: ".$token."<br>";

// Create document
$doc = array('path' => '/okm:root/hosts.txt', 'mimeType' => null,
    'actualVersion' => null, 'author' => null, 'checkedOut' => false,
    'created' => null, 'keywords' => 'test', 'language' => null,
    'lastModified' => null, 'lockInfo' => null, 'locked' => false,
    'permissions' => 0, 'size' => 0, 'subscribed' => false, 'uuid' => null,
    'convertibleToPdf' => false, 'convertibleToSwf' => false,
    'compactable' => false, 'training' => false, 'convertibleToDxf' => false,
    'signed' => false, 'cipherName' => null, 'subscriptors'=>null, 'title'=>null,
    'description'=>null, 'categories'=>null, 'notes'=>null);

$createResp = $OKMDocument->create(array('token' => $token, 'doc' => $doc, 'content' => $newDoc));
$newDoc = $createResp->return;
echo "[DOCUMENT] Path: ".$newDoc->path.", Author: ".$newDoc->author.", Size: ".$newDoc->size;

// Logout
$OKMAuth->logout(array('token' => $token));
?>
```



We encourage using `createSimple` method rather `create` method.

About document variables during creation process only will be take on consideration the "**path2**" variable, the others will be filled internally during creation process.

Create document and set a category

```
<?php
// Register WSDL
$OKMAuth = new SoapClient('http://localhost:8080/OpenKM/services/OKMAuth?wsdl');
$OKMDocument = new SoapClient('http://localhost:8080/OpenKM/services/OKMDocument?wsdl');
$OKMProperty = new SoapClient('http://localhost:8080/OpenKM/services/OKMProperty?wsdl');
$OKMRepository = new SoapClient('http://localhost:8080/OpenKM/services/OKMRepository?wsdl');
$file = '/etc/hosts';

// Login
$loginResp = $OKMAuth->login(array('user' => 'okmAdmin', 'password' => 'admin'));
$token = $loginResp->return;
echo "Token: ".$token."<br>";

// Create document
$createResp = $OKMDocument->createSimple(array('token' => $token, 'docPath' => '/okm:root/hosts.txt', 'content' => $newDoc));
$newDoc = $createResp->return;
echo "[DOCUMENT] Path: ".$newDoc->path.", Author: ".$newDoc->author.", Size: ".$newDoc->size;

// Category assign
$getNodeUuid = $OKMRepository->getNodeUuid(array('token' => $token, 'path' => '/okm:root/hosts.txt'));
$catId = $getNodeUuid->return;
$OKMProperty->addCategory(array('token' => $token, 'nodeId' => $newDoc->path, 'categoryId' => $catId));
?>
```

```
// Logout
$OKMAuth->logout($token);
?>
```

Search by content

```
<?php
// Register WSDL
$OKMAuth = new SoapClient('http://localhost:8080/OpenKM/services/OKMAuth?wsdl');
$OKMSearch = new SoapClient('http://localhost:8080/OpenKM/services/OKMSearch?wsdl');

// Login
$loginResp = $OKMAuth->login(array('user' => 'okmAdmin', 'password' => 'admin'));
$token = $loginResp->return;
echo "Token: ".$token."<br>";

$findByContentResp = $OKMSearch->findByContent(array('token' => $token, 'content' =>
if (isset($getChildrenResp->return)) {
    $queryResultArray = $findByContentResp->return;

    if ($queryResultArray) {
        if (is_array($queryResultArray)) {
            foreach ($queryResultArray as $queryResult) {
                echo "-> ".$queryResult->document->path." (".$queryResult->score.")<br>";
            }
        } else {
            echo "-> ".$queryResultArray->document->path." (".$queryResultArray->score.")
        }
    }
}

// Logout
$OKMAuth->logout(array('token' => $token));
?>
```

Search with parameters

```
<?php
// Register WSDL
$OKMAuth = new SoapClient('http://localhost:8080/OpenKM/services/OKMAuth?wsdl');
$OKMSearch = new SoapClient('http://localhost:8080/OpenKM/services/OKMSearch?wsdl');

// Login
$loginResp = $OKMAuth->login(array('user' => 'okmAdmin', 'password' => 'admin'));
$token = $loginResp->return;
echo "Token: ".$token."<br>\n";

$qqp = new QueryParams();
$qp->domain = 1; // DOCUMENT = 1; FOLDER = 2; MAIL = 4; DOCUMENT & FOLDER = 3; etc.
$qp->content = 'test';
$findResp = $OKMSearch->find(array('token' => $token, 'params' => $qp));
if (isset($getChildrenResp->return)) {
    $queryResultArray = $findResp->return;

    if ($queryResultArray) {
        if (is_array($queryResultArray)) {
            foreach ($queryResultArray as $queryResult) {
                echo "-> ".$queryResult->document->path." (".$queryResult->score.")<br>\n";
            }
        } else {
            echo "-> ".$queryResultArray->document->path." (".$queryResultArray->score.")
        }
    }
}
```

```

    }
  }
}

// Logout
$OKMAuth->logout(array('token' => $token));

class QueryParams {
    var $content = '';
    var $dashboard = false;
    var $domain = 0;
    var $id = 0;
    var $properties = array();
}
?>

```

Set metadata group values

The metadata group definition used:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.0//EN"
    "http://www.openkm.com/dtd/property-groups-2.0.dtd">
<property-groups>
  <property-group label="Technology" name="okp:technology">
    <select label="Type" name="okp:technology.type" type="multiple">
      <option label="Alfa" value="t1"/>
      <option label="Beta" value="t2" />
      <option label="Omega" value="t3" />
    </select>
    <select label="Language" name="okp:technology.language" type="simple">
      <option label="Java" value="java"/>
      <option label="Python" value="python"/>
      <option label="PHP" value="php" />
    </select>
    <input label="Comment" name="okp:technology.comment"/>
    <textarea label="Description" name="okp:technology.description"/>
    <input label="Link" type="link" name="okp:technology.link"/>
  </property-group>
</property-groups>

```



The node must already have set the group, otherwise you'll get an error when trying to set values of a non previous added metadata group.

```

<?php
$OKMAuth = new SoapClient('http://localhost:8080/OpenKM/services/OKMAuth?wsdl');
$OKMPropertyGroup = new SoapClient('http://localhost:8080/OpenKM/services/OKMPropertyGroup');

$loginResp = $OKMAuth->login(array('user' => 'okmAdmin', 'password' => 'admin'));
$token = $loginResp->return;
echo "Token: ".$token;

$docPath = "/okm:root/hosts.txt";
$entry['key'] = 'okp:technology.comment';
$entry['value'] = 'Other comment from PHP 3';
$properties = array($entry);
$OKMPropertyGroup->setPropertiesSimple(array('token' => $token, 'nodeId' => $docPath));
$OKMAuth->logout($token);
?>

```

Exception handling*The "finally" has been introduced from PHP version 5.5.*

```

<?php
// Register WSDL
$OKMAuth = new SoapClient('http://localhost:8080/OpenKM/services/OKMAuth?wsdl');
$OKMDocument = new SoapClient('http://localhost:8080/OpenKM/services/OKMDocument?wsdl');

function format_exception($e) {
    if (isset($e->detail)) {
        $reflectionObject = new ReflectionObject($e->detail);
        $properties = $reflectionObject->getProperties();
        $exceptionName = $properties[0]->name;
    } else {
        $exceptionName = "Exception";
    }
    return $exceptionName . ": " . $e->faultstring;
}

try {
    $loginResp = $OKMAuth->login(array('user' => 'okmAdmin', 'password' => 'admin'));
    $token = $loginResp->return;
    $getPropertiesResp = $OKMDocument->getProperties(array('token' => $token, 'docPath' => ''));
    $docProps = $getPropertiesResp->return;
    print_r($docProps);
} catch (Exception $e) {
    echo format_exception($e);
}

$OKMAuth->logout(array('token' => $token));
?>

```

Proxy configuration

```

$client = new SoapClient("some.wsdl", array('proxy_host' => "https://example.org", 'proxy_port' => 443));

```

Extending frontend UI



To extend frontend you need to get the source code and compile the changes into.

The OpenKM plugin extensions makes it easy to extend the OpenKM end user interface by encapsulating plugin code and making it reusable between OpenKM versions.

The OpenKM plugin extension architecture is based on:

- Extensions
- Events
- Handlers
- Communicators

Extensions are available widget definitions that allows the developer to make extensible panels and widgets (for example adding new tab panel on tab document)

Events are a collection of events that OpenKM UI triggers each time any changes occur (for example when a new keyword is added in the tab document the triggered event `HasDocumentEvent.KEYWORD_ADDED`)

Handlers are a collection of methods called internally by OpenKM . Handlers must be implemented into your extensions in order to collect OpenKM triggered events. Each extension you make can have one or several handlers, that are automatically registered by OpenKM on the loading process. OpenKM internally triggers events to each declared handler.

Communicators are a collection of methods available as OpenKM Comunitators API to access transparently some internal UI values. There are several Communicators, for example with `GeneralComunicator` you can access some general actions like refreshing UI as `GeneralComunicator.refreshUI()`. OpenKM Communicators API has read and write methods to interact with internal OpenKM UI objects.



Creating OpenKM plugin extensions is easy but you need some java knowledge and pay special attention to the Google Web ToolKit API that's used to build OpenKM UI. <http://code.google.com/webtoolkit/>

Enable sample extensions

OpenKM comes with some frontend UI sample extensions not enabled by default. To enable them go to **Administration > Utilities** > [Database query](#) and execute the SQL queries:

Extension	SQL sentence
ToolBarButtonExample	<pre>INSERT INTO OKM_EXTENSION (EXT_UUID,EXT_NAME) VALUES ('9f84b330-d</pre>

TabWorkspaceExample	<pre>INSERT INTO OKM_EXTENSION (EXT_UUID,EXT_NAME) VALUES('44f94470-d</pre>
TabFolderExample	<pre>INSERT INTO OKM_EXTENSION (EXT_UUID,EXT_NAME) VALUES('d95e01a0-d</pre>
TabFolderExample	<pre>INSERT INTO OKM_EXTENSION (EXT_UUID,EXT_NAME) VALUES('d9dab640-d</pre>



After registering a sample extension, you also need to enable the from **Administration** > **Profiles**.

Frontend UI hello world sample

Description



This example add a new Widget in tab documents user interface:

- Create a new GWT module.
- Import new GWT module into default OpenKM frontend GWT module.
- Create a new basic widget.
- Use widget into OpenKM frontend UI.

Sample

Create a new GWT module

Create a file called **HelloWorld.gwt.xml** into **src/main/resources/com/openkm/extension/frontend**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE module SYSTEM "http://google-web-toolkit.googlecode.com/svn/releases/2.0/di
<module>
  <!-- Inherit the core Web Toolkit stuff -->
  <inherits name='com.google.gwt.user.User' />
  <inherits name="com.google.gwt.http.HTTP" />
</module>
```

Add module into OpenKM frontend UI

Edit **Customization.gwt.xml** into **src/main/resources/com/openkm/extension/frontend** and add the new module

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE module SYSTEM "http://google-web-toolkit.googlecode.com/svn/releases/2.0/di
<module>
  <!-- Inherit the extension widgets -->
  <inherits name="com.openkm.extension.frontend.HelloWorld" />
</module>
```

Create new widget

Create a file called **HelloWorld.java** into **src/main/java/com/openkm/extension/frontend/client**

```
public class HelloWorld extends TabDocumentExtension {
    Button refresh;
    VerticalPanel vPanel;

    public HelloWorld() {
        HTML html = new HTML("Hello Word");
        refresh = new Button("refresh UI");
        refresh.addClickHandler(new ClickHandler() {
            @Override
```

```

        public void onClick(ClickEvent event) {
            GeneralCommunicator.refreshUI();
        }
    });
    vPanel = new VerticalPanel();
    vPanel.add(html);
    vPanel.add(refresh);

    refresh.setStyleName("okm-Input");

    initWidget(vPanel);
}

@Override
public String getTabText() {
    return "Hello tab";
}

@Override
public void set(GWTDocument doc) {
    // TODO Auto-generated method stub
}

@Override
public void setVisibleButtons(boolean visible) {
    // TODO Auto-generated method stub
}

    @Override
    public String getExtensionUUID() {
        return String.valueOf("d9dab640-d098-11df-bd3b-0800200c9a66");
    }
}

```

Use widget into OpenKM

Edit **Customization.java** OpenKM class and add the HelloWorld widget:

```

public class Customization {

    /**
     * getExtensionWidgets
     *
     * @return
     */
    public static List<Object> getExtensionWidgets() {
        List<Object> extensions = new ArrayList<Object>();

        // Declare here your widget extensions
        extensions.add(new HelloWorld());

        return extensions;
    }
}

```

Results

Compile project, login into OpenKM and you'll see something like this:







Frontend UI extensions

Extensions are available widget definitions that allows the developer to make extensible panels and widgets (for example adding new tab panel on tab document).

Available extensions sections are:

Widget	Workspace	Description
Workspace	Desktop	<p>Location: Top right.</p> <p>Usually they are desktop, search, dashboard and administrator tabs.</p> <div>  More information at <i>TabWorkspaceExtension</i> . </div>
Tab document.	Desktop	<p>Location: Bottom right.</p> <p>Usually they are properties, notes, security, history, and preview tabs.</p> <div>  More information at <i>TabDocumentExtension</i> . </div>
Tab folder	Desktop	<p>Location: Bottom right.</p> <p>Usually they are properties, notes and security tabs.</p> <div>  More information at <i>TabFolderExtension</i> . </div>
Tab mail	Desktop	<p>Location: Bottom right.</p> <p>Usually they are properties, notes, security and preview tabs.</p> <div>  More information at <i>TabMailExtension</i> . </div>
Toolbar box	Dashboard	<p>Location: Dashboard toolbar.</p> <p>Toolbar icons</p>

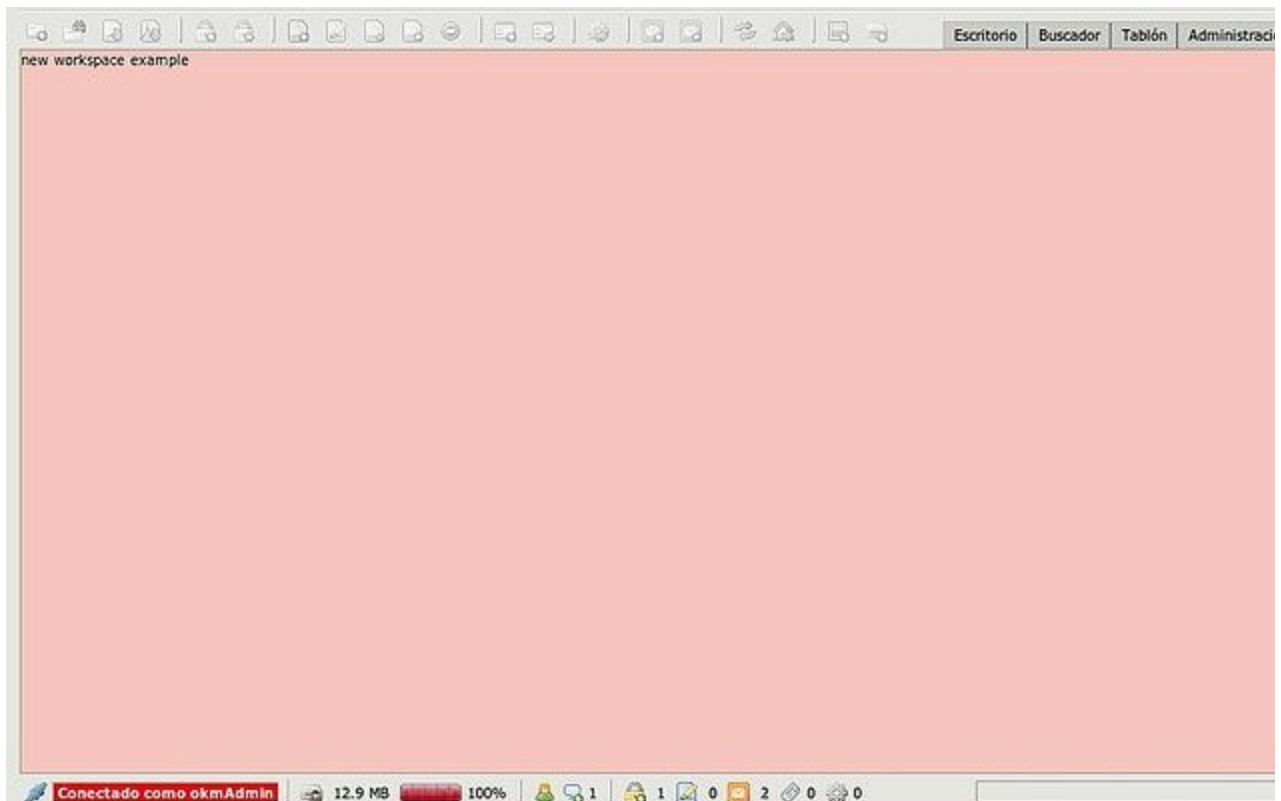
		 More information at ToolBarBoxExtension .
Toolbar button	Desktop	<p>Location: Top center.</p> <p>Toolbar icons of the OpenKM features like create folder, delete, upload new file, etc.</p>  More information at ToolBarButtonExtension .
Menu Bar	Desktop	<p>Location: Top.</p> <p>Usually the menu bar options are file, edit, tools, bookmarks, templates and help menus.</p>  More information at MenuBarExtension .
Menu Item	Desktop	<p>Location: Top.</p> <p>Here are set the menu options of each main menu bar.</p>  More information at MenuItemExtension .

Widget library

Tab workspace



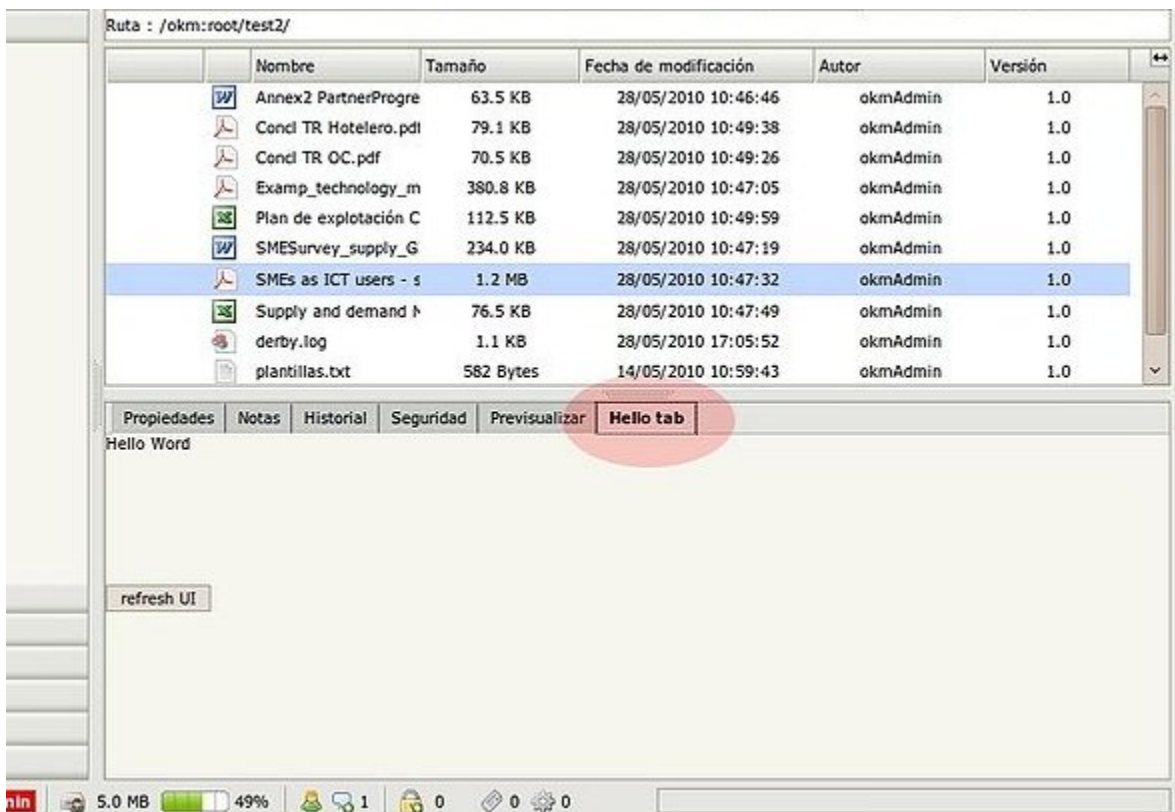
More information at: [TabWorkspaceExtension](#).



Tab Document



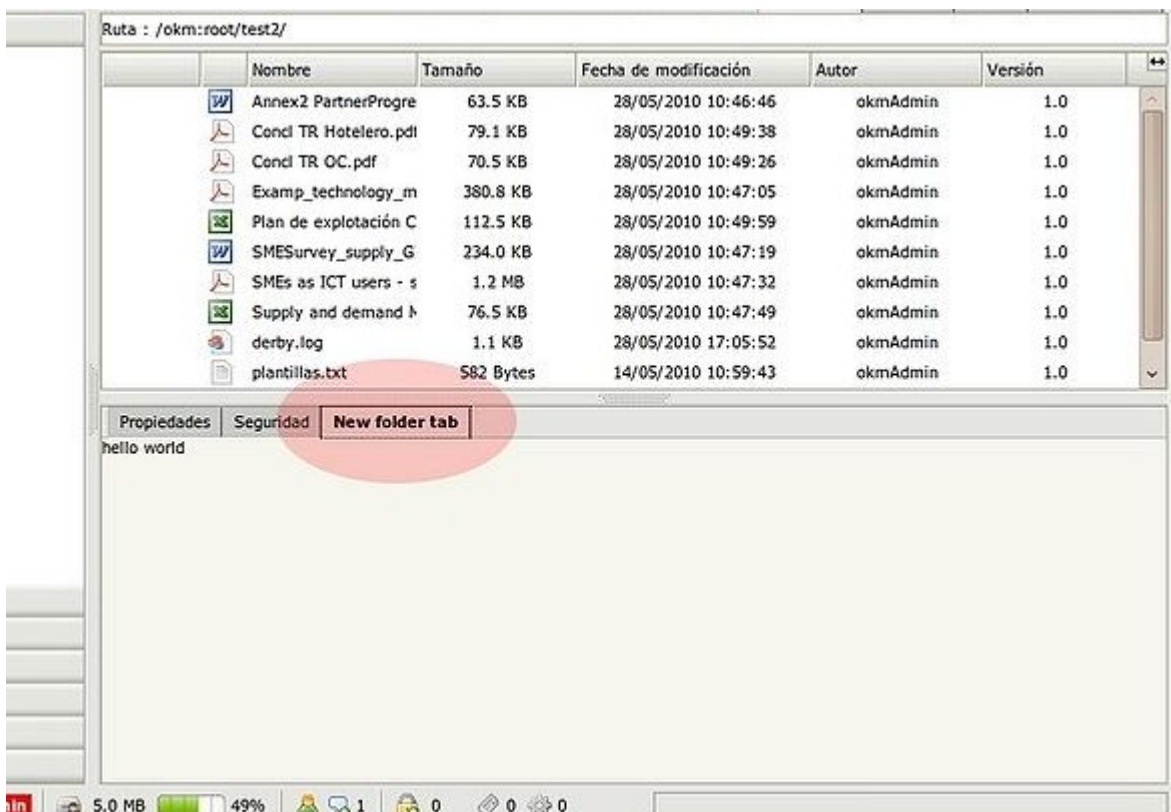
More information at: [taTabDocumentExtension](#).



Tab folder



More information at: [TabFolderExtension](#).



Main menu



More information at: [MenuBarExtension](#) and [MenuItemExtension](#).



Toolbar button



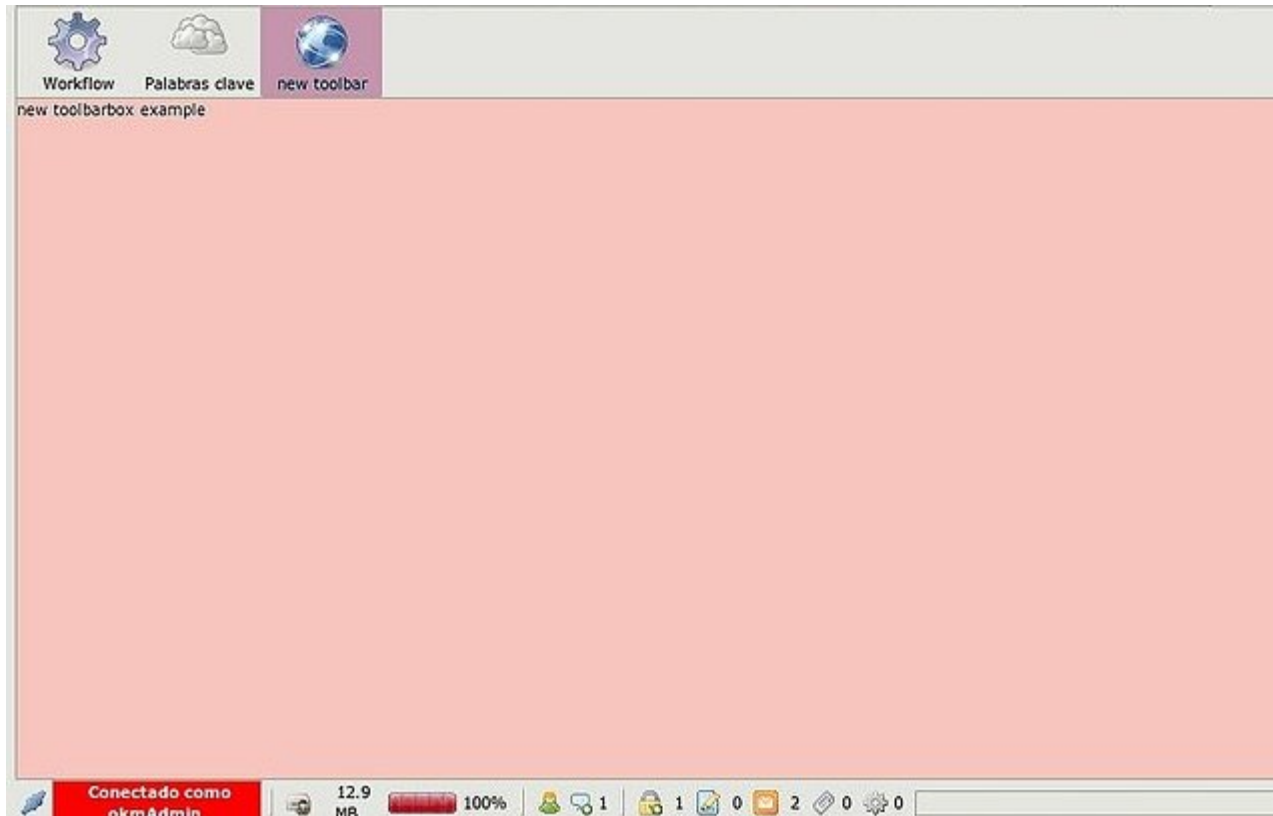
More information at: [ToolBarButtonExtension](#).



Dashboard toolbar button



More information at: [ToolBarBoxExtension](#).



TabWorkspace Extension



Adds a new tab at top right workspace section (usually there are shown desktop, search, dashboard and administration tab).

Methods

method	Description
getTabText()	Used by OpenKM to get the tab text.
setTab(TabBar tabBar, int tabIndex)	Used by openkm to set tabBar and tab index.
getExtensionUUID()	Return the unique extension id

Basic sample

```
public class TabWorkspaceExample extends TabWorkspaceExtension {
    private VerticalPanel vPanel;

    /**
     * TabWorkspaceExample
     */
    public TabWorkspaceExample() {
        vPanel = new VerticalPanel();
        vPanel.add(new HTML("new workspace example"));

        vPanel.setStyleName("okm-Input");

        initWidget(vPanel);
    }

    @Override
    public String getTabText() {
        return "tab workspace";
    }

    @Override
    public void setTab(TabBar tabBar, int tabIndex) {
    }

    @Override
    public String getExtensionUUID() {
        return String.valueOf("44f94470-d097-11df-bd3b-0800200c9a66");
    }
}
```

Sample with iframe

```

public class TabWorkspace extends TabWorkspaceExtension {
    private VerticalPanel vPanel;
    private Frame iframe;
    private String textLabel = "";
    private TabBar tabBar;
    private int tabIndex = 0;

    /**
     * TabWorkspace
     */
    public TabWorkspace() {
        vPanel = new VerticalPanel();
        iframe = new Frame("about:blank");

        DOM.setElementProperty(iframe.getElement(), "frameborder", "0");
        DOM.setElementProperty(iframe.getElement(), "marginwidth", "0");
        DOM.setElementProperty(iframe.getElement(), "marginheight", "0");

        // Commented because on IE show clear if allowtransparency=true
        DOM.setElementProperty(iframe.getElement(), "allowtransparency", "false");
        DOM.setElementProperty(iframe.getElement(), "scrolling", "no");

        iframe.setUrl(Main.CONTEXT + "/sample/index.jsp");
        iframe.setStyleName("okm-Iframe");

        vPanel.add(iframe);
        vPanel.setCellHorizontalAlignment(iframe, HasAlignment.ALIGN_CENTER);

        vPanel.setWidth("100%");
        vPanel.setHeight("100%");

        initWidget(vPanel);
    }

    /**
     * Sets the size on initialization
     *
     * @param width The max width of the widget
     * @param height The max height of the widget
     */
    public void setPixelSize(int width, int height) {
        iframe.setPixelSize(width - 2, height - 2);
    }

    /**
     * setTextLabel
     */
    public void setTextLabel(String textLabel) {
        this.textLabel = textLabel;
    }

    @Override
    public String getTabText() {
        return textLabel;
    }

    @Override
    public void setTab(TabBar tabBar, int tabIndex) {
        this.tabBar = tabBar;
        this.tabIndex = tabIndex;
    }

    @Override
    public String getExtensionUUID() {
        return String.valueOf("44f94470-d097-11df-bd3b-0800200c9a66");
    }
}

```

```
}
```

WorkspaceHandlerExtension



Any extension that implements *WorkspaceHandlerExtension* will be looking at all the events fired from the workspace tab panel.

Methods

method	Description
<i>onChange(WorkspaceEventConstant event)</i>	Each time it'll be some new tab panel event the method <i>onChange</i> it'll be triggered by OpenKM.

Sample

```
public class HandlersTest implements WorkspaceHandlerExtension {
    @Override
    public void onChange(WorkspaceEventConstant event) {
        Window.alert("workspace event fired");
    }
}
```

WorkspaceCommunicator



Use it to retrieve workspace data.

Methods

method	return	type	Description
<i>getSelectedTab()</i>	<i>int</i>	<i>read</i>	<i>Get the selected tab id.</i>
<i>changeSelectedTab(int selectedTab)</i>	<i>void</i>	<i>write</i>	<i>Change the selected tab.</i>
<i>getSelectedWorkspace()</i>	<i>int</i>	<i>read</i>	<i>Get selected workspace id.</i>
<i>getWorkspace()</i>	<i>GWTWorkspace</i>	<i>read</i>	<i>Get selected workspace object.</i>
<i>getTabExtensionIndex(TabWorkspaceExtension widget)</i>	<i>int</i>	<i>read</i>	<i>Get the tab id of the widget.</i>
<i>getAppVersion()</i>	<i>String</i>	<i>read</i>	<i>Retrieve the application version.</i>

Sample

```
int tab = WorkspaceCommunicator.getSelectedTab();
if (tab==0) {
    Window.alert("First tab enabled");
} else {
    Window.alert("Other stack");
}
```

TabDocument Extension



Add a new tab at bottom right (usually there are shown properties, notes, security, history, and preview tabs) in the document tab section.

Methods

method	Description
getTabText()	Used by OpenKM to get the tab text.
set(GWTDocument doc)	Each time there's some document selected (in browser) is executed this method by OpenKM to refreshing document tab panel information.
setVisibleButtons(boolean visible)	Shows or hides buttons (for example in trash view usually edit button should not be visible).

Basic sample

```
public class TabDocumentExample extends TabDocumentExtension {
    VerticalPanel vPanel;
    String tabText = "Tab example";

    public TabDocumentExample() {
        HTML html = new HTML("Content tab example");
        vPanel = new VerticalPanel();
        vPanel.add(html);

        initWidget(vPanel);
    }

    @Override
    public String getTabText() {
        return tabText;
    }
}
```

DocumentHandlerExtension



Any extension that implements *DocumentHandlerExtension* will look at all the events triggered from the document tab.

Methods

method	Description
<i>onChange(DocumentEventConstant event)</i>	Each time there is a new document event, the method <i>onChange</i> will be triggered by OpenKM .

Sample

```
public class TabDocumentExample extends TabDocumentExtension implements DocumentHandler {
    VerticalPanel vPanel;
    String tabText = "Tab - example";

    public TabDocumentExample() {
        HTML html = new HTML("Content tab example");
        vPanel = new VerticalPanel();
        vPanel.add(html);

        initWidget(vPanel);
    }

    @Override
    public String getTabText() {
        return tabText;
    }

    @Override
    public void set(GWTDocument doc) {
        // TODO Auto-generated method stub
    }

    @Override
    public void setVisibleButtons(boolean visible) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onChange(DocumentEventConstant event) {
        if (event.equals(HasDocumentEvent.DOCUMENT_CHANGED)) {
            Window.alert("document changed - " + event.getType());
        } else if (event.equals(HasDocumentEvent.KEYWORD_ADDED)) {
            Window.alert("keyword added - " + event.getType());
        } else if (event.equals(HasDocumentEvent.KEYWORD_REMOVED)) {
            Window.alert("keyword removed - " + event.getType());
        } else if (event.equals(HasDocumentEvent.CATEGORY_ADDED)) {
            Window.alert("category added - " + event.getType());
        } else if (event.equals(HasDocumentEvent.CATEGORY_REMOVED)) {
            Window.alert("category removed - " + event.getType());
        } else if (event.equals(HasDocumentEvent.TAB_CHANGED)) {

```



```
Window.alert("tab changed - " +event.getType() + " - actual tab " + TabDocumentCom
} else if (event.equals(HasDocumentEvent.PANEL_RESIZED)) {
Window.alert("panel resized - " +event.getType());
} else if (event.equals(HasDocumentEvent.SECURITY_CHANGED)) {
Window.alert("security changed - " +event.getType());
} else if (event.equals(HasDocumentEvent.NOTE_ADDED)) {
Window.alert("note added - " +event.getType());
}
}
}
```

TabDocumentComunicator



Use to retrieve tab document data.

Methods

method	return	type	Description
<i>getSelectedTab()</i>	<i>int</i>	<i>read</i>	Get the selected tab id.
<i>getDocument()</i>	<i>GWTDocument</i>	<i>read</i>	Retrieve the GWT Document object from selected document of the user interface.
<i>getKeywords()</i>	<i>Collection<String></i>	<i>read</i>	Retrieve a list of keywords from selected document of the user interface.
<i>Collection<GWTNote></i>	<i>getNotes()</i>	<i>read</i>	Retrieve a list of notes from selected document of the user interface.
<i>isVisibleButton()</i>	<i>boolean</i>	<i>read</i>	Return true when buttons are enabled on this view.
<i>refreshPreviewDocument()</i>	<i>void</i>	<i>read</i>	Force refreshing preview tab.
<i>isWidgetExtensionVisible(Widget widget)</i>	<i>boolean</i>	<i>read</i>	Return true when the widget is visible in document tabs.
<i>addKeyword(String keyword)</i>	<i>void</i>	<i>write</i>	Add a keyword to the selected document of the user interface.
<i>removeKeyword(String keyword)</i>	<i>void</i>	<i>write</i>	Delete a keyword to the selected document of the user interface.
<i>addCategory(GWTFolder category)</i>	<i>void</i>	<i>write</i>	Add a category to the selected document of the user interface.
<i>removeCategory(String uuid)</i>	<i>void</i>	<i>write</i>	Delete a category to the selected document of the user interface.

<i>setRefreshingStyle()</i>	<i>void</i>	<i>write</i>	<i>Change panel style to refreshing value.</i>
<i>unsetRefreshingStyle()</i>	<i>void</i>	<i>write</i>	<i>Change panel style to default value.</i>

Sample

```
int selectedTab = TabDocumentComunicator.getSelectedTab();
if (selectedTab==0) {
    Window.alert("First tab enabled");
} else {
    Window.alert("Other tab");
}
```

TabFolderExtension



Add a new tab at bottom right (usually there are shown properties, notes and security tabs) in folder tab section.

Methods

method	Description
getTabText()	Used by OpenKM to get the tab text.
set(GWTFolder folder)	Each time some folder is selected (at tree or browser) is executed this method by OpenKM to refreshing folder tab panel information.
setVisibleButtons(boolean visible)	Show or hide buttons (for example in trash view usually edit button should not be visible).

Sample

```
public class TabFolderExample extends TabFolderExtension {
    VerticalPanel vPanel;

    public TabFolderExample() {
        vPanel = new VerticalPanel();
        vPanel.add(new HTML("hello world"));

        initWidget(vPanel);
    }

    @Override
    public String getTabText() {
        return "New folder tab";
    }

    @Override
    public void set(GWTFolder folder) {
        // TODO Auto-generated method stub
    }

    @Override
    public void setVisibleButtons(boolean visible) {
        // TODO Auto-generated method stub
    }
}
```

FolderHandlerExtension



Any extension that implements *FolderHandlerExtension* will be looking all events triggered from folder ta.

Methods

method	Description
<i>onChange(FolderEventConstant event)</i>	Each time it'll be some new folder event the method <i>onChange</i> it'll be fired by OpenKM .

Sample

```
public class TabFolderExample extends TabFolderExtension implements FolderHandlerExtension {
    VerticalPanel vPanel;

    public TabFolderExample() {
        vPanel = new VerticalPanel();
        vPanel.add(new HTML("hello world"));

        initWidget(vPanel);
    }

    @Override
    public String getTabText() {
        return "New folder tab";
    }

    @Override
    public void set(GWTFolder doc) {
        // TODO Auto-generated method stub
    }

    @Override
    public void setVisibleButtons(boolean visible) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onChange(FolderEventConstant event) {
        if (event.equals(HasFolderEvent.TAB_CHANGED)) {
            Window.alert("tab changed - " + event.getType());
        }
    }
}
```

TabFolder Comunicator



Use to retrieve tab folder data.

Methods

method	return	type	Description
<i>getSelectedTab()</i>	<i>int</i>	<i>read</i>	<i>Gets the selected tab id.</i>
<i>getFolder()</i>	<i>GWTFolder</i>	<i>read</i>	<i>Retrieves the GWT folder object from the selected folder of the user interface.</i>
<i>isVisibleButton()</i>	<i>boolean</i>	<i>read</i>	<i>Returns true when buttons are enabled on this view.</i>
<i>isWidgetExtensionVisible(Widget widget)</i>	<i>boolean</i>	<i>read</i>	<i>Returns true when the widget is visible in folder tabs.</i>

Sample

```
int selectedTab = TabFolderComunicator.getSelectedTab();
if (selectedTab==0) {
    Window.alert("First tab enabled");
} else {
    Window.alert("Other tab");
}
```

TabMail Extension



Adds a new tab at bottom right (usually there are shown properties, notes, security and preview tabs) in mail tab section.

Methods

method	Description
<code>getTabText()</code>	Used by OpenKM to get the tab text.
<code>set(GWTMail mail)</code>	Each time there's a mail selected (in a browser) it is executed this method by OpenKM to refresh mail tab panel information.
<code>setVisibleButtons(boolean visible)</code>	Shows or hide buttons (for example in trash view usually edit button should not be visible).

Sample

```
public class TabMailExample extends TabMailExtension {
    VerticalPanel vPanel;

    public TabMailExample() {
        vPanel = new VerticalPanel();
        vPanel.add(new HTML("hello world"));

        initWidget(vPanel);
    }

    @Override
    public String getTabText() {
        return "New folder tab";
    }

    @Override
    public void set(GWTMail mail) {
        // TODO Auto-generated method stub
    }

    @Override
    public void setVisibleButtons(boolean visible) {
        // TODO Auto-generated method stub
    }
}
```

MailHandlerExtension



Any extension that implements MailHandlerExtension will be looking all events triggered from mail ta.

Methods

method	Description
onChange(MailEventConstant event)	Each time it'll be some new mail event the method onChange it'll be fired by OpenKM.

Sample

```
public class TabMailExample extends TabMailExtension implements MailHandlerExtension {
    VerticalPanel vPanel;

    public TabMailExample() {
        vPanel = new VerticalPanel();
        vPanel.add(new HTML("hello world"));

        initWidget(vPanel);
    }

    @Override
    public String getTabText() {
        return "New folder tab";
    }

    @Override
    public void set(GWTMail mail) {
        // TODO Auto-generated method stub
    }

    @Override
    public void setVisibleButtons(boolean visible) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onChange(MailEventConstant event) {
        if (event.equals(HasMailEvent.TAB_CHANGED)) {
            Window.alert("tab changed - " + event.getType());
        }
    }
}
```


TabMail Comunicator



Used to retrieve tab mail data.

Methods

method	return	type	Description
<i>getSelectedTab()</i>	<i>int</i>	<i>read</i>	Gets the selected tab id.
<i>getMail()</i>	<i>GWTMail</i>	<i>read</i>	Retrieves the GWT mail object from selected mail of the user interface.
<i>isWidgetExtensionVisible(Widget widget)</i>	<i>boolean</i>	<i>read</i>	Returns true when the widget is visible in mail tabs.

Sample

```
int selectedTab = TabMailComunicator.getSelectedTab();
if (selectedTab==0) {
    Window.alert("First tab enabled");
} else {
    Window.alert("Other tab");
}
```

ToolBarBox Extension



Adds a new button at the dashboard view.

Methods

method	Description
<i>getWidget()</i>	Returns the associated widget to the tool box that will be showed when it'll is selected.
<i>getExtensionUUID()</i>	Return the unique extension id

Sample

```
public class ToolBarBoxExample extends ToolBarBoxExtension {
    public ToolBarBoxEx(Image img, String text) {
        super(img, text);
    }

    @Override
    public Widget getWidget() {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public String getExtensionUUID() {
        return String.valueOf("d9dab640-d098-11df-bd3b-0800200c9a66");
    }
}
```

DashboardHandlerExtension



Any extension that implements `ToolBarHandlerExtension` will be monitoring events triggered from the toolbar.

Methods

method	Description
<code>onChange(DashboardEventConstant event)</code>	Each time it'll be some new toolbar event the method <code>onChange</code> it'll be fired by OpenKM .

Sample

```
package com.openkm.extension.frontend.client;

import com.google.gwt.user.client.Window;
import com.google.gwt.user.client.ui.HTML;
import com.google.gwt.user.client.ui.HorizontalPanel;
import com.google.gwt.user.client.ui.Image;
import com.google.gwt.user.client.ui.Widget;
import com.openkm.extension.frontend.client.util.OKMExtensionBundleExampleResources;
import com.openkm.frontend.client.extension.event.HasDashboardEvent;
import com.openkm.frontend.client.extension.event.HasToolBarEvent;
import com.openkm.frontend.client.extension.event.handler.ToolBarHandlerExtension;
import com.openkm.frontend.client.extension.widget.toolbar.ToolBarBoxExtension;

public class ToolBarBoxExample implements DashboardHandlerExtension {

    public HorizontalPanel hPanel;
    public ToolBarBoxExtension toolBarBoxExtension;

    /**
     * ToolBarBoxExample
     */
    public ToolBarBoxExample() {
        toolBarBoxExtension = new ToolBarBoxExtension(new Image(OKMExtensionBundleExampleResources.getResource("new_toolbar.png")), "new toolbar") {
            @Override
            public Widget getWidget() {
                return hPanel;
            }
        };

        hPanel = new HorizontalPanel();
        hPanel.add(new HTML("new toolbarbox example"));
    }

    /**
     * ToolBarBoxExtension
     */
    public ToolBarBoxExtension getToolBarBox() {
        return toolBarBoxExtension;
    }

    @Override
```

```
public void onChange(DashboardEventConstant event) {  
    if (event.equals(HasDashboardEvent.TOOLBOX_CHANGED)) {  
        Window.alert("toolbar changed - " + event.getType());  
    }  
}
```

ToolBarComunicator



Use it to retrieve tab dashboard data.

Methods

method	return	type	Description
<i>getUserSubscribedDocuments()</i>	<i>void</i>	<i>read</i>	Refresh user subscribed documents list.
<i>getUserSubscribedFolders()</i>	<i>void</i>	<i>read</i>	Refresh user subscribed folders list.
<i>refreshAllSearchs()</i>	<i>void</i>	<i>read</i>	Refresh search list.
<i>showToolBoxExtension(ToolBarBoxExtension extension)</i>	<i>void</i>	<i>read</i>	Change selected dashboard button.
<i>getUserSearchs(boolean refresh)</i>	<i>void</i>	<i>read</i>	Refresh user search list. <div> i When refresh boolean value is true, also refresh search results at dashboard news view. </div>

Sample

```
// Call to refreshing subscribed documents
DashboardComunicator.getUserSubscribedDocuments();
```

ToolBarButton Extension



Adds a new toolbar icon. Usually toolbar icons are OpenKM features like create folder, delete, upload new file, etc.

Methods

method	Description
<i>checkPermissions(GWTNode node, GWTNode nodeParent, int originPanel)</i>	Checks the button permissions based on the selected node grants to enable or disable the button.
<i>checkPermissions(GWTDocument doc, GWTNode node)</i>	Checks the button permissions based on selected document grants to enable or disable the button.
<i>checkPermissions(GWTMail mail, GWTNode node)</i>	Checks the button permissions based on the selected folder grants to enable or disable the button.
<i>checkPermissions(GWTRecord record, GWTNode node);</i>	Checks the button permissions based on the selected record node grants to enable or disable the button.
<i>isEnabled()</i>	Returns a boolean value indicating if the button is enabled or disabled.
<i>enable(boolean enable)</i>	Enables or disables button.
<i>getExtensionUUID()</i>	Returns the unique extension id

Sample

```
public class ToolBarButton extends ToolBarButtonExtension {

    public ToolBarButton(Image image, String title, ClickHandler handler) {
        super(image, title, handler);
    }

    @Override
    public void checkPermissions(GWTNode node, GWTNode nodeParent, int originPanel) {
        // TODO Auto-generated method stub
    }
}
```

```
}

@Override
public void checkPermissions(GWTDocument doc, GWTNode node) {
    // TODO Auto-generated method stub
}

@Override
public void checkPermissions(GWTMail mail, GWTNode node) {
    // TODO Auto-generated method stub
}

@Override
public void checkPermissions(GWTRecord record, GWTNode node) {
    // TODO Auto-generated method stub
}

@Override
public void enable(boolean enable) {
    // TODO Auto-generated method stub
}

@Override
public boolean isEnabled() {
    // TODO Auto-generated method stub
    return false;
}

@Override
public String getExtensionUUID() {
    return "9f84b330-d096-11df-bd3b-0800200c9a66";
}
}
```

ToolBarHandler Extension



Any extension that implements `ToolBarHandlerExtension` will be monitoring events triggered from the toolbar.

Methods

method	Description
<code>onChange(ToolBarEventConstant event)</code>	Each time it'll be some new toolbar event the method <code>onChange</code> it'll be fired by OpenKM .

Sample

```
public class ToolBarButton extends ToolBarButtonExtension implements ToolBarHandlerEx

public ToolBarButton(Image image, String title, ClickHandler handler) {
    super(image, title, handler);
}

@Override
public void checkPermissions(GWTNode node, GWTNode folderParent, int originPanel) {
    // TODO Auto-generated method stub
}

@Override
public void checkPermissions(GWTDocument doc, GWTNode node) {
    // TODO Auto-generated method stub
}

@Override
public void checkPermissions(GWTMail mail, GWTNode node) {
    // TODO Auto-generated method stub
}

@Override
public void checkPermissions(GWTRecord record, GWTNode node) {
    // TODO Auto-generated method stub
}

@Override
public void enable(boolean enable) {
    // TODO Auto-generated method stub
}

@Override
public boolean isEnabled() {
    // TODO Auto-generated method stub
    return false;
}

@Override
public void onChange(ToolBarEventConstant event) {
    if (event.equals(HasToolBarEvent.EXECUTE_ADD_DOCUMENT)) {
        Window.alert("executed add document - " + event.getType());
    }
}
```



```
}  
}
```

MenuBarExtension



Add new menu bar.

Constructor

method	Description
MenuBarExtension()	Create a new menu bar or submenu.

Sample

```
public class MainMenuExample {
    private MenuItemExtension exampleMenu;
    private MenuBarExtension subMenuExample;
    private MenuBarExtension subMenuExample2;
    private MenuItemExtension subMenuItem;
    private MenuItemExtension option1;
    private MenuItemExtension option2;
    private MenuItemExtension option3;
    private MenuItemExtension option4;

    /**
     * MainMenuExample
     */
    public MainMenuExample() {
        // All menu items
        option1 = new MenuItemExtension("img/box.png", "Option 1", option1Action);
        option2 = new MenuItemExtension("img/box.png", "Option 2", option2Action);
        option3 = new MenuItemExtension("img/box.png", "Option 3", option3Action);
        option4 = new MenuItemExtension("img/box.png", "Option 4", option4Action);

        // Secondary submenu
        subMenuExample2 = new MenuBarExtension();
        subMenuExample2.addItem(option3);
        subMenuExample2.addItem(option4);
        subMenuItem = new MenuItemExtension("img/box.png", "Sub menu", subMenuExample2);

        // Principal submenu
        subMenuExample = new MenuBarExtension();
        subMenuExample.addItem(option1);
        subMenuExample.addItem(option2);
        subMenuExample.addItem(subMenuItem);

        // Principal menuitem
        exampleMenu = new MenuItemExtension("New Menu", subMenuExample); // is not a
    }

    public MenuItemExtension getNewMenu() {
        return exampleMenu;
    }

    /**
     * option1Action
     */
    Command option1Action = new Command() {
```

```
        public void execute() {
            Window.alert("option1 action");
        }
    };

    /**
     * option2Action
     */
    Command option2Action = new Command() {
        public void execute() {
            Window.alert("option2 action");
        }
    };

    /**
     * option3Action
     */
    Command option3Action = new Command() {
        public void execute() {
            Window.alert("option3 action");
        }
    };

    /**
     * option4Action
     */
    Command option4Action = new Command() {
        public void execute() {
            Window.alert("option4 action");
        }
    };
}
```

MenuItemExtension



Add new menu option to the menu bar.

Constructors

method	Description
MenuItemExtension(String imageURL, String text, Command cmd)	Default menuItem.
MenuItemExtension(String imageURL, String text, MenuBar menuBar)	A menuItem that has a menu bar into. Usually used for a sub menu entry.
MenuItemExtension(String text, MenuBar menuBar)	Usually used to define the main menuItem. This is the root menu that has menu children options.

Sample

```
public class MainMenuExample {
    private MenuItemExtension exampleMenu;
    private MenuBarExtension subMenuExample;
    private MenuBarExtension subMenuExample2;
    private MenuItemExtension subMenuItem;
    private MenuItemExtension option1;
    private MenuItemExtension option2;
    private MenuItemExtension option3;
    private MenuItemExtension option4;

    /**
     * MainMenuExample
     */
    public MainMenuExample() {
        // All menu items
        option1 = new MenuItemExtension("img/box.png", "Option 1", option1Action);
        option2 = new MenuItemExtension("img/box.png", "Option 2", option2Action);
        option3 = new MenuItemExtension("img/box.png", "Option 3", option3Action);
        option4 = new MenuItemExtension("img/box.png", "Option 4", option4Action);

        // Secondary submenu
        subMenuExample2 = new MenuBarExtension();
        subMenuExample2.addItem(option3);
        subMenuExample2.addItem(option4);
        subMenuItem = new MenuItemExtension("img/box.png", "Sub menu", subMenuExample2); //

        // Principal submenu
        subMenuExample = new MenuBarExtension();
        subMenuExample.addItem(option1);
        subMenuExample.addItem(option2);
    }
}
```

```
subMenuExample.addItem(subMenuItem);

// Principal menuitem
exampleMenu = new MenuItemExtension("New Menu", subMenuExample); // is not a second
}

public MenuItemExtension getNewMenu() {
    return exampleMenu;
}

/**
 * option1Action
 */
Command option1Action = new Command() {
    public void execute() {
        Window.alert("option1 action");
    }
};

/**
 * option2Action
 */
Command option2Action = new Command() {
    public void execute() {
        Window.alert("option2 action");
    }
};

/**
 * option3Action
 */
Command option3Action = new Command() {
    public void execute() {
        Window.alert("option3 action");
    }
};

/**
 * option4Action
 */
Command option4Action = new Command() {
    public void execute() {
        Window.alert("option4 action");
    }
};
}
```

General HandlerExtension stuff

NavigatorHandlerExtension



Any extension that implements NavigatorHandlerExtension will be watching all events fired from navigator stack panel.

Methods

method	Description
onChange(NavigatorEventConstant event)	Each time it'll be a new stack panel event the method onChange it'll be fired by OpenKM.

Sample

```
public class HandlersTest implements NavigatorHandlerExtension {
    @Override
    public void onChange(NavigatorEventConstant event) {
        Window.alert("navigator event fired");
    }
}
```

PropertyGroupHandlerExtension



Any extension that implements PropertyGroupHandlerExtension will be watching all events fired from metadata group tab.

Methods

method	Description
onChange(PropertyGroupEventConstant event)	Each time it'll be a new property group (metadata) event the method onChange it'll be fired by OpenKM.

Sample

```
public class Example implements PropertyHandlerExtension {

    public Example() {
    }

    @Override
    public void onChange(PropertyEventConstant event) {
        if (event.equals(HasPropertyGroupEvent.EVENT_PROPERTYGROUP_CHANGED)) {
        }
    }
}
```

```

        Window.alert("propertyGroup changed - " +event.getType());
    } else {
        Window.alert("other property group event - " +event.getType());
    }
}

```

WidgetHandlerExtension



Any extension that implements `WidgetHandlerExtension` will be watching all events fired from widgets.

Methods

method	Description
<code>onChange(WidgetEventConstant event)</code>	Each time it'll be a new widget event the method <code>onChange</code> it'll be fired by OpenKM .

Sample

```

public class Example implements WidgetHandlerExtension {

    public Example() {
    }

    @Override
    public void onChange(WidgetEventConstant event) {
        if (event.equals(HasWidgetEvent.FINISH_STARTUP)) {
            // Do something here
        }
    }
}

```





General Comunicator stuff

GeneralComunicator





Use it to retrieve general data.

Methods

method	return	type	Description
refreshUI()	void	read	Refresh user interface.
getToolBarOption()	ToolBarOption	read	Retrive ToolbarOption object. <div>  <p>The variables into ToolbarOption shown OpenKM features enabled or disabled by security.</p> </div>
getLang()	String	read	Get user interface language code.
downloadDocument(boolean checkout)	void	read	Download selected document. <div>  <p>When checkout value is "true" the document will be marked as "edition".</p> </div>
downloadDocumentPdf()	void	read	Download selected document as pdf.
downloadFile(String path, String params)	void	read	Download file with parameters. <div>  <p>Deprecated method in favritesr "downloadFileByUUID" method.</p> </div>
downloadFileByUUID(String uuid, String params)	void	read	Download file with parameters. <div>  <p>Allowed parameters:</p> </div>

			<ul style="list-style-type: none"> • The "checkout" parameter mark document as "edit". • The "inline" parameter to download as attachment, for example to downloading images. • The "ver" parameter to download specific document version.
<i>extensionCallOwnDownload(String url)</i>	<i>void</i>	<i>read</i>	Used to call
<i>resetStatus()</i>	<i>void</i>	<i>read</i>	Reset bottom right status bar.
<i>showError(String callback, Throwable caught)</i>	<i>void</i>	<i>read</i>	Show popup error with some message.
<i>logout()</i>	<i>void</i>	<i>read</i>	Logout from OpenKM user interface.
<i>refreshUserDocumentsSize()</i>	<i>void</i>	<i>read</i>	Refresh user documents size.
<i>getUserRoleList()</i>	<i>List<String></i>	<i>read</i>	Retrieve a list with the roles of the user.
<i>getUser()</i>	<i>String</i>	<i>read</i>	Retrieve userId.
<i>i18nExtension(String property)</i>	<i>String</i>	<i>read</i>	Retrieve extension translation.
<i>openPath(String path, String docPath)</i>	<i>void</i>	<i>read</i>	Open node into openkm. <div> i Parameters: <ul style="list-style-type: none"> • Parameter "path" should be a valid "folder path". </div>

			<ul style="list-style-type: none"> Parameter "docPath" should be empty or a valid "document path". <div>  To open a single folder path the parameter "docPath" must be empty. </div>
getAppContext()	String	read	Retrieve the application context.
showNextWizard()	void	read	Show next wizard. <div>  Wizards are shown after document uploading. </div>
getWorkspace()	GWTWorkspace	read	Retrieve the GWT Workspace object. <div>  Into the object is set information about UI features - enabled or disabled - among others. </div>
i18n(String property)	String	read	Retrieve frontend translation.
enableKeyShorcuts()	void	read	Enable UI shortcuts.
disableKeyShorcuts()	void	void	Disable UI shortcuts.
openPathByUuid(String uuid)	void	void	Open node into openkm.
getFolderIcon(GWTFolder fld)	String	void	Retrieve folder icon image path.
get()	Main	void	Retrieve Main object from singleton pattern. <div>Main class is the principal frontend</div>

			 <i>class which offers full access to any frontend UI method.</i>
<i>setStatus(String msg)</i>	<i>void</i>	<i>void</i>	On bottom right change status message.

Sample

```
// Call to refreshing user interface
GeneralCommunicator.refreshUI();
```

NavigatorComunicator


Use it to retrieve navigator data.



The navigator panel contains stack panel with folder trees into (taxonomy, categories etc...).

Methods

<i>method</i>	<i>return</i>	<i>type</i>	<i>Description</i>
<i>getStackIndex()</i>	<i>int</i>	<i>read</i>	Get the stack index id.
<i>isTaxonomyShown()</i>	<i>boolean</i>	<i>read</i>	When true, indicate the taxonomy stack is shown.
<i>isCategoriesShown()</i>	<i>boolean</i>	<i>read</i>	When true, indicate the categories stack is shown.
<i>isMetadataShown()</i>	<i>boolean</i>	<i>read</i>	When true, indicate the metadata stack is shown.
<i>isThesaurusShown()</i>	<i>boolean</i>	<i>read</i>	When true, indicate the thesaurus stack is shown.
<i>isTemplatesShown</i>	<i>boolean</i>	<i>read</i>	When true, indicate the templates stack is shown.
<i>isPersonalShown</i>	<i>boolean</i>	<i>read</i>	When true, indicate the personal stack is shown.
<i>isMailShown</i>	<i>boolean</i>	<i>read</i>	When true, indicate the mail stack is shown.

<i>isTrashShown</i>	<i>boolean</i>	<i>read</i>	When true, indicate the trash stack is shown.
<i>getFolder()</i>	<i>GWTFolder</i>	<i>read</i>	Retrieve the selected folder object from enabled tree view. <div style="border: 1px dashed orange; padding: 5px; background-color: #fff9c4; display: inline-block;">  We suggest using <i>getNode()</i> method rather this one. </div>
<i>getActualPath()</i>	<i>String</i>	<i>read</i>	Retrieve the path of the selected folder from enabled tree view.
<i>getRootFolder()</i>	<i>GWTFolder</i>	<i>read</i>	Retrieve the GWT folder object of node <i>"/okm:root"</i> .
<i>getCategoriesRootFolder()</i>	<i>GWTFolder</i>	<i>read</i>	Retrieve the GWT folder object of node <i>"/okm:categories"</i> .
<i>getMetadataRootFolder()</i>	<i>GWTFolder</i>	<i>read</i>	Retrieve the GWT folder object of node <i>"/okm:metadata"</i> .
<i>getThesaurusRootFolder()</i>	<i>GWTFolder</i>	<i>read</i>	Retrieve the GWT folder object of node <i>"/okm:thesaurus"</i> .
<i>getTemplatesRootFolder()</i>	<i>GWTFolder</i>	<i>read</i>	Retrieve the GWT folder object of node <i>"/okm:templates"</i> .
<i>getPersonalRootFolder()</i>	<i>GWTFolder</i>	<i>read</i>	Retrieve the GWT folder object of node <i>"/okm:personal"</i> .
<i>getMailRootFolder()</i>	<i>GWTFolder</i>	<i>read</i>	Retrieve the GWT folder object of node <i>"/okm:mail"</i> .
<i>getTrashRootFolder()</i>	<i>GWTFolder</i>	<i>read</i>	Retrieve the GWT folder object of node <i>"/okm:trash"</i> .

Sample

```
int stack = NavigatorComunicator.getStackIndex();
if (stack==0) {
    Window.alert("First stack enabled");
} else {
    Window.alert("Other stack");
}
```

FileBrowserComunicator



Use it to retrieve file browser data.

Methods

method	return	type	Description
<i>isDocumentSelected()</i>	<i>boolean</i>	<i>read</i>	When document is selected it returns "true" value.
<i>isFolderSelected()</i>	<i>boolean</i>	<i>read</i>	When folder is selected it returns "true" value.
<i>isMailSelected()</i>	<i>boolean</i>	<i>read</i>	When mail is selected it returns "true" value.
<i>getDocument()</i>	<i>GWTDocument</i>	<i>read</i>	Retrieve a GWT document object of the selected document.
<i>getFolder()</i>	<i>GWTFolder</i>	<i>read</i>	Retrieve a GWT folder object of the selected folder.
<i>getMail()</i>	<i>GWTMail</i>	<i>read</i>	Retrieve a GWT mail object of the selected mail.
<i>isPanelSelected()</i>	<i>boolean</i>	<i>read</i>	When file browser panel is selected return "true" value.
<i>refreshOnlyFileBrowser()</i>	<i>void</i>	<i>read</i>	Refresh only file browser widget view.

Sample

```
if (FileBrowserCommunicator.isDocumentSelected()) {
    Window.alert("document selected");
} else {
    Window.alert("document not selected");
}
```



UtilCommunicator




General purpose methods.

Methods

method	return	type	Description

<i>formatSize(double size)</i>	<i>String</i>	<i>read</i>	Convert document size in bytes to human value (Kb, Mb, GB etc..).
<i>imageItemHTML(String imageUrl, String title, String align)</i>	<i>String</i>	<i>read</i>	Retrieve image HTML code to be used into GWT frontend widget.
<i>createHeaderHTML(String imageURL, String caption)</i>	<i>String</i>	<i>read</i>	Retrieve a stack header HTML code to be used into GWT frontend stack widget.
<i>menuHTML(String imageUrl, String text)</i>	<i>String</i>	<i>read</i>	Retrieve menu HTML code to be used into GWT frontend menu element.
<i>imageItemHTML(String imageUrl)</i>	<i>String</i>	<i>read</i>	Retrieve image HTML code to be used into GWT frontend widget.
<i>getTextAsBoldHTML(String text, boolean mark)</i>	<i>String</i>	<i>read</i>	Retrieve text as HTML bold one.
<i>getUserAgent()</i>	<i>String</i>	<i>read</i>	Retrieve user agent of the browser.
<i>getParent(String path)</i>	<i>String</i>	<i>read</i>	Retrieve the parent path of a path. <div>  The parent of the "/okm:root/temp/doc1.pdf" is "/okm:root/temp". </div>
<i>getName(String path)</i>	<i>String</i>	<i>read</i>	Retrieve the name of a path. <div>  The name of the "/okm:root/temp/doc1.pdf" is "doc1.pdf". </div>
<i>mimeImageHTML(String mime)</i>	<i>String</i>	<i>read</i>	Retrieve the url of the MIME TYPE icon.
<i>hSpace(String width)</i>	<i>HTML</i>	<i>read</i>	Retrieve a GWT HTML horizontal separator.

vSpace(String height)	HTML	read	Retrieve a GWT HTML vertical separator.
isSearchableKey(KeyUpEvent event)	boolean	read	Evaluate if the key pressed is not into navigation, arrows or modifier keys. <div>  <p>Navigation keys: HOME, END, PAGEUP, PAGEDOWN.</p> <p>Modifier keys: SHIFT, ALT, CTRL, DOWN.</p> <p>Arrow keys: LEFT, RIGHT, UP, DOWN.</p> </div>
menuHTMLWithoutIcon(String text)	String	read	Retrieve menu HTML code to be used into GWT frontend menu element.

Sample

```
// Call to formatting the document size ( gb, mb etc... )
UtilComunicator.formatSize(15);
```

SearchComunicator

Use it to retrieve search data.

Methods

method	return	type	Description
getAllSearchs()	void	read	Refresh saved search panel.
getUserSearchs()	void	read	Refresh user saved search panel.
getSelectedRowSearchSaved()	int	read	Retrieve the selected row of saved search panel.
getSelectedRowUserNews()	int	read	Retrieve the selected row of user saved search panel.
			Retrieve the GWT QueryParams object of the

<i>getSavedSearch()</i>	<i>GWTQueryParams</i>	<i>read</i>	<i>selected saved search.</i>
<i>getSavedUserNews()</i>	<i>GWTQueryParams</i>	<i>read</i>	<i>Retrieve the GWT QueryParams object of the selected user saved search.</i>
<i>setSavedSearch(GWTQueryParams params)</i>	<i>void</i>	<i>write</i>	<i>Execute a saved search.</i>

Sample

```
// Call to refreshing subscribed documents
GWTQueryParams params = new GWTQueryParams();
params.setName("test");
SearchCommunicator.setSavedSearch(params);
```

ToolBarCommunicator

Use it to retrieve toolbar data.

Methods

<i>method</i>	<i>return</i>	<i>type</i>	<i>Description</i>
<i>evaluateShowIcons()</i>	<i>void</i>	<i>read</i>	<i>Evaluate toolbar icons based on security.</i>
<i>getToolBarOption()</i>	<i>ToolBarOption</i>	<i>read</i>	<i>Retrieve the GWT ToolBarOption object what contains information about enabled or disabled UI features.</i>
<i>getActualNode()</i>	<i>Object</i>	<i>read</i>	<i>Retrieve the selected node of the user interface.</i> <div> i Retrieved object can be cast to: <ul style="list-style-type: none"> • <i>GWTDocument.</i> • <i>GWTFolder.</i> • <i>GWTMail.</i> • <i>GWTRecrod.</i> </div>

<i>getMainToolBarPanel()</i>	<i>HorizontalPanel</i>	<i>read</i>	<i>Retrieve the main widget of the tool bar.</i>
<i>setToolBarOption(ToolBarOption toolBarOption)</i>	<i>void</i>	<i>write</i>	<i>Change toolbar option, it enable or disable UI features.</i>

Sample

```
ToolBarCommunicator.evaluateShowIcons() ;
```

UI constants

UIMenuConstants



UIMenuConstant class set constants to be used in OpenKM extensions, for example to set new menu options on some existing OpenKM menu.



*When menu location is not set by default is used value **NEW_MENU**. That means it'll be created as new menu into main menu desktop view.*

Sample

Add new submenu in tools menu:

```
public class SubMenuMessage {
    private MenuItemExtension messageMenu;
    private MenuBarExtension subMenuMessage;
    private MenuItemExtension sendNewMessage;

    /**
     * SubMenuMessage
     */
    public SubMenuMessage() {
        // All menu items
        sendNewMessage = new MenuItemExtension("img/icon/actions/new_message.png", "New mes

        // Principal submenu
        subMenuMessage = new MenuBarExtension();
        subMenuMessage.addItem(sendNewMessage);
        messageMenu = new MenuItemExtension("img/icon/actions/message.png", "Message", subM
        messageMenu.setMenuLocation(UIMenuConstants.MAIN_MENU_TOOLS);
    }

    /**
     * @return
     */
    public MenuItemExtension getMenu() {
        return messageMenu;
    }

    /**
     * option1Action
     */
    Command sendMessage = new Command() {
        public void execute() {
            Window.alert("some action");
        }
    };
}
```

UIDesktopConstants



Set constants to be used in OpenKM extensions to identify some desktop widgets.

Sample

```
if (NavigatorComunicator.getStackIndex()==UIDesktopConstants.NAVIGATOR_TAXONOMY) {  
    Window.alert("Taxonomy selected");  
} else {  
    Window.alert("Other navigator panel selected");  
}
```

UIDockPanelConstants

Set contants to be used in OpenKM extensions to identify some dock panel widgets (main widgets).

Sample

```
WorkspaceComunicator.changeSelectedTab(UIDockPanelConstants.SEARCH);
```

UISearchConstants

Set contants to be used in OpenKM extensions to identify some widgets on search view.

UIFileUploadConstants

Set contants to be used in OpenKM extensions to identify some operation on fileupload.

Frontend UI extensions stuff

Internatinalization

Register new translations from [Database query](#).

```
INSERT INTO OKM_TRANSLATION (TR_MODULE, TR_KEY, TR_TEXT, TR_LANGUAGE) VALUES ('extens
INSERT INTO OKM_TRANSLATION (TR_MODULE, TR_KEY, TR_TEXT, TR_LANGUAGE) VALUES ('extens
```

Use in your code:

```
GeneralComunicator.il8nExtension("new_extension.button.add");
```

Also can be used default OpenKM translations:

```
GeneralComunicator.il8n("button.close");
```

Sample



Remember any extension internationalized must implements **LanguageHandlerExtension**. Each time the user interface change language this handler is fired.

```
private class ToolBarButton extends ToolBarButtonExtension implements LanguageHandlerExtension {
    public ToolBarButton(Image image, String title, ClickHandler handler) {
        super(image, title, handler);
    }

    @Override
    public void checkPermissions(GWTNode node, GWTNode nodeParent, int originPanel) {
        // TODO
    }

    @Override
    public void checkPermissions(GWTDocument doc, GWTNode node) {
        // TODO
    }

    @Override
    public void checkPermissions(GWTMail mail, GWTNode node) {
        // TODO
    }

    @Override
    public void checkPermissions(GWTRecord record, GWTNode node) {
        // TODO
    }

    @Override
    public void enable(boolean enable) {
        // TODO
    }
}
```

```

@Override
public boolean isEnabled() {
    return true;
}

@Override
public void onChange(LanguageEventConstant event) {
    if (event.equals(HasLanguageEvent.LANGUAGE_CHANGED)) {
        setTitle(GeneralComunicator.i18n("download.button.tittle"));
    }
}
}

```

Registering RPC (Remote Process Call) service

Servlets are used to do GWT RPC (Remote Process Calls). In order to registering a new Servlet file (web.xml) must be modified and OpenKM re-compiled.

Into **web.xml** (src/main/webapp/WEB-INF) must be added the servlet declaration:

```

<servlet>
    <servlet-name>SomeServletName</servlet-name>
    <servlet-class>SomeJavaClassName</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>SomeServletName</servlet-name>
    <url-pattern>/SomeServletName</url-pattern>
</servlet-mapping>

```

RPCService class defines all RPC constants service that can be used from GWT:

```

ServiceDefTarget endPoint = (ServiceDefTarget) messageService;
endPoint.setServiceEntryPoint(RPCService.MessageService);
messageService.deleteSent(1, new AsyncCallback<Object>() {
    @Override
    public void onSuccess(Object result) {
    }

    @Override
    public void onFailure(Throwable caught) {
    }
});

```



More information at <http://www.gwtproject.org/doc/latest/tutorial/RPC.html>.

Complete frontend extension sample

This sample shows how to quickly extend OpenKM features with JSP, HTML, CSS and JavaScript technologies using OpenKM Java API and how to use JavaScript functions of the exposed frontend JAVA methods.

Download the extension sample code [complete_extension_sample.zip](#).

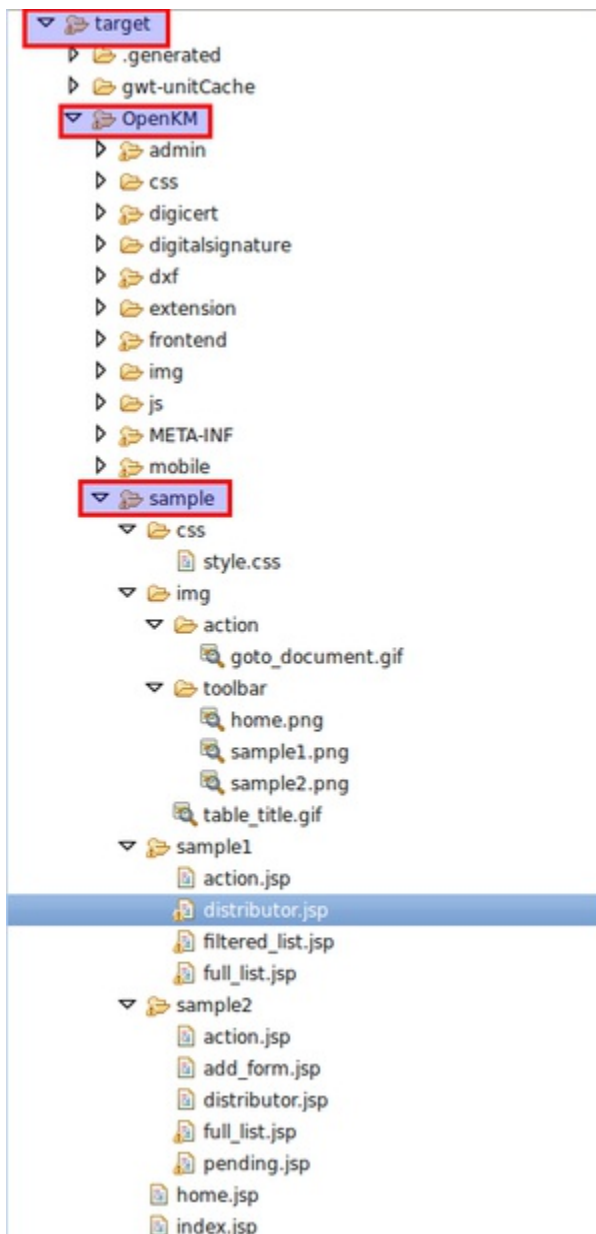
Description

The sample is shown into a new Workspace tab.

The main screen is shown as a tool bar with tree buttons:

- *The first button goes to main screen of the sample.*
- *The second button show sample 1*
- *The second button show sample 2.*

Project structure:



Installation

- Stop the application.
- Open the content of the download file into \$TOMCAT_HOME/webapps/OpenKM folder (or better into OpenKM.war)
- Delete \$TOMCAT_HOME/work/Catalina/localhost
- Start the application.
- Check the url <http://localhost:8080/OpenKM/sample>.



The screen shots has been done in combination with "**ExtraTab**" frontend extension feature, also could be used [TabWorkspaceExtension](#).

When you change the contents of the OpenKM.war the file is deployed as a new version of the application and all

the contents into \$TOMCAT_HOME/webapps/OpenKM will be refreshed and you will lost any change you made directly in the webapps/OpenKM folder.

Implementation details

index.jsp

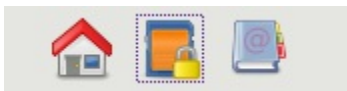
Index.jsp control the iframe height with a JavaScript code.

*Other important fact is, for example after uploading a document (Sample 2), is used the parameter **urlToOpen** to indicate which url must be opened.*

```
String urlToOpen = URLDecoder.decode(WebUtils.getString(request, "urlToOpen"), "UTF-8")
if (urlToOpen.equals("")) {
    urlToOpen = "home.jsp";
}
```

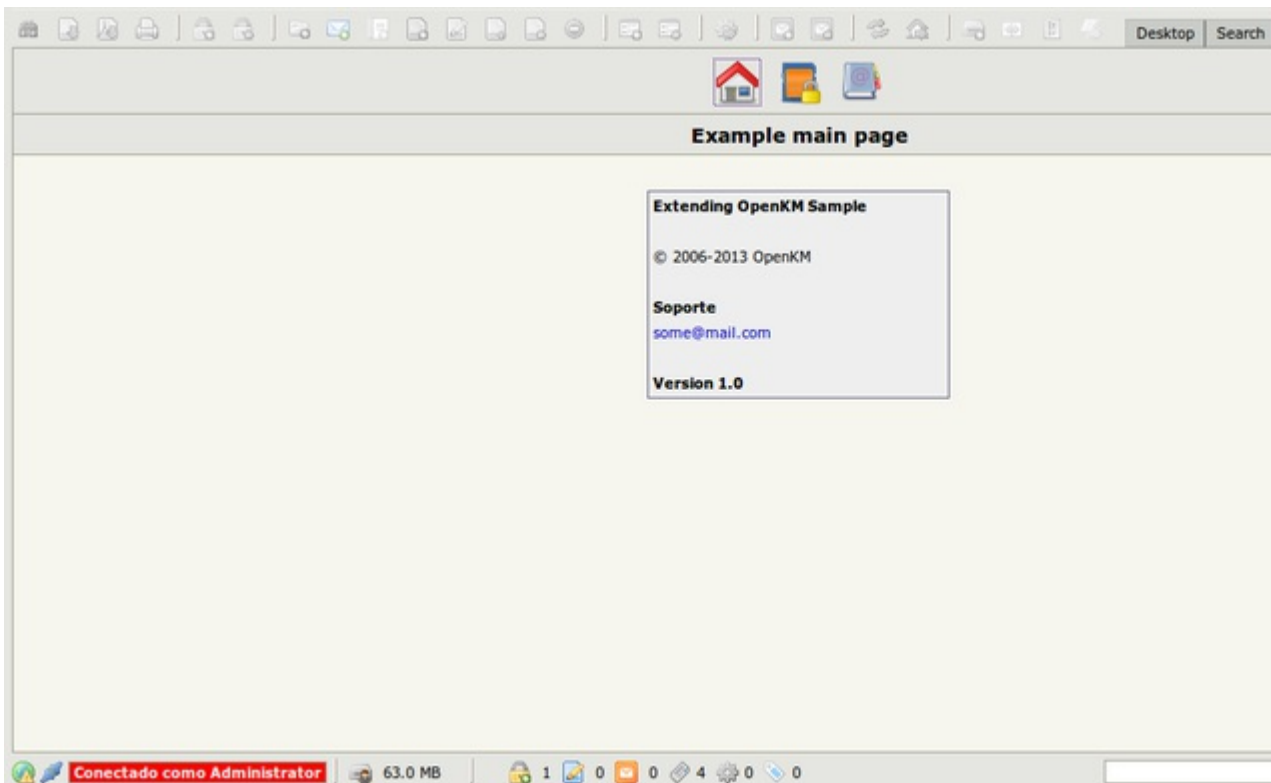
menu.jsp

Has the menu url definitions.



home.jsp

Set the main home screen.



Sample 1

Is composed by 4 files:

- *action.jsp* (the main controller where is set the logic).
- *distributor.jsp* (the main layer with two columns at 50%).
- *full_list.jsp* (show full list with no JavaScript special control).
- *filtered_list.jsp* (show filtered list with no JavaScript special control).

action.jsp

The main controller.

```
<%@page import="java.util.Map"%>
<%@page import="java.util.HashMap"%>
<%@page import="java.util.List"%>
<%@page import="java.util.ArrayList"%>
<%@ page import="java.text.SimpleDateFormat" %>
<%@ page import="java.io.IOException" %>
<%@ page import="org.slf4j.Logger" %>
<%@ page import="org.slf4j.LoggerFactory" %>
<%@ page import="com.openkm.bean.Document" %>
<%@ page import="com.openkm.util.WebUtils" %>
<%@ page import="com.openkm.api.OKMDocument" %>
<%@ page extends="com.openkm.extension.servlet.BaseServlet" %>
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%!
private static Logger log = LoggerFactory.getLogger("com.openkm.sample");
```

```

public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
    ServletException {
        String action = WebUtils.getString(request, "action");
        try {
            list(request, response);
        } catch (Exception e) {
            log.error(e.getMessage(), e);
            sendErrorRedirect(request, response, e);
        }
    }
}

// Used by fileupload return and filter
public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException {
    String action = WebUtils.getString(request, "action");
    try {
        list(request, response);
    } catch (Exception e) {
        log.error(e.getMessage(), e);
        sendErrorRedirect(request, response, e);
    }
}

private void list(HttpServletRequest request, HttpServletResponse response) throws ServletException {
    SimpleDateFormat df = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
    String docPath = (String) request.getAttribute("docPath");
    String nameFilter = WebUtils.getString(request, "nameFilter");

    // Pending docs
    List<Map<String,String>> pendingDocMaps = new ArrayList<Map<String,String>>();
    for ( Document doc : OKMDocument.getInstance().getChildren(null, "/okm:root") ) {
        Map<String,String> docMap = new HashMap<String,String>();
        docMap.put("name", doc.getPath().substring(doc.getPath().lastIndexOf("/") + 1));
        docMap.put("uuid", doc.getUuid());
        docMap.put("path", doc.getPath());
        docMap.put("mimeType", doc.getMimeType());
        docMap.put("lastModified", df.format(doc.getLastModified().getTime()));
        pendingDocMaps.add(docMap);
    }

    ServletContext sc = getServletContext();
    sc.setAttribute("docPath", docPath);
    sc.setAttribute("nameFilter", nameFilter);
    sc.setAttribute("pendingDocuments", pendingDocMaps);
    sc.getRequestDispatcher("/sample/sample2/distributor.jsp").forward(request, response);
}

```

distributor.jsp

The main layer.

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <link rel="stylesheet" type="text/css" href="<%=request.getContextPath() %>/sample/sample2/css/main.css" />
    <script type="text/javascript" src="<%=request.getContextPath() %>/js/jquery-1.7.1.min.js" />
    <script type="text/javascript" src="<%=request.getContextPath() %>/js/vanadium-min.js" />
</head>
<body>
    <table width="100%">
        <tbody>

```

```

<tr>
  <td valign="top" width="50%">
    <jsp:include page="/sample/sample1/full_list.jsp"></jsp:include>
  </td>
  <td valign="top" width="10%"></td>
  <td valign="top" width="50%">
    <jsp:include page="/sample/sample1/filtered_list.jsp"></jsp:include>
  </td>
</tr>
</tbody>
</table>
</body>
</html>

```

full_list.jsp

Show document list.



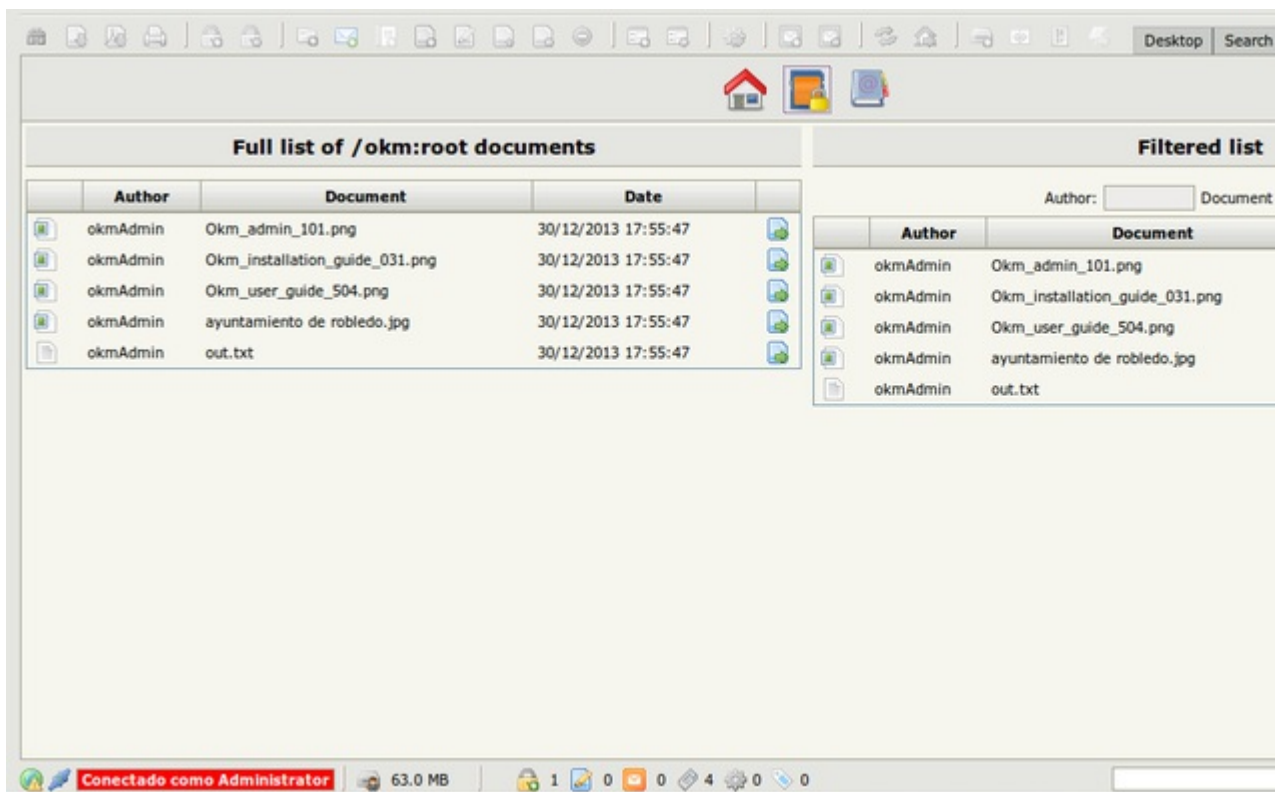
Pay attention at JavaScript function "**go**", which call `parent.parent.jsOpenPathByUuid(uuid)`. The "**jsOpenPathByUuid**" is a GWT public function used to jump to some specific document, folder, etc. Here [Exposed frontend JavaScript API](#) there's a complete list of JavaScript functions.

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
<script type="text/javascript">
  // Note the parent.parent call because iframe is into other iframe
  function go(uuid) {
    alert('Document UUID: '+uuid);
    parent.parent.jsOpenPathByUuid(uuid);
  }
</script>
<h1>Full list of /okm:root documents</h1>
<table class="results" style="white-space: nowrap;" cellpadding="3" width="100%">
  <thead>
    <tr>
      <th></th><th>Author</th><th>Document</th><th>Date</th><th></th>
    </tr>
  </thead>
  <tbody>
    <c:forEach var="document" items="${leftDocsMaps}" varStatus="row">
      <tr>
        <td valign="center">
        <td>${document.author}</td>
        <td>${document.name}</td>
        <td>${document.lastModified}</td>
        <td align="center">
          <a href="javascript:go('${document.uuid}')">
        </td>
      </tr>
    </c:forEach>
  </tbody>
</table>

```

Screenshot of sample1 view



Sample 2

Is composed by 4 files:

- *action.jsp* (the main controller where is set the logic).
- *add_form.jsp* (form to upload a new document)
- *distributor.jsp* (the main layer with two columns at 50% with JavaScript).
- *full_list.jsp* (show full list with JavaScript special control for table height).
- *pending.jsp* (static content).

action.jsp

The main controller.

```
<%@page import="java.util.Map"%>
<%@page import="java.util.HashMap"%>
<%@page import="java.util.List"%>
<%@page import="java.util.ArrayList"%>
<%@ page import="java.text.SimpleDateFormat" %>
<%@ page import="java.io.IOException" %>
<%@ page import="org.slf4j.Logger" %>
<%@ page import="org.slf4j.LoggerFactory" %>
<%@ page import="com.openkm.bean.Document" %>
<%@ page import="com.openkm.util.WebUtils" %>
<%@ page import="com.openkm.api.OKMDocument" %>
```

```

<%@ page extends="com.openkm.extension.servlet.BaseServlet" %>
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>

private static Logger log = LoggerFactory.getLogger("com.openkm.sample");
public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
    ServletException {
        String action = WebUtils.getString(request, "action");
        try {
            list(request, response);
        } catch (Exception e) {
            log.error(e.getMessage(), e);
            sendErrorRedirect(request, response, e);
        }
    }

    // Used by fileupload return and filter
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException {
        ServletException {
            String action = WebUtils.getString(request, "action");
            try {
                list(request, response);
            } catch (Exception e) {
                log.error(e.getMessage(), e);
                sendErrorRedirect(request, response, e);
            }
        }
    }

    private void list(HttpServletRequest request, HttpServletResponse response) throws ServletException {
        SimpleDateFormat df = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
        String docPath = (String) request.getAttribute("docPath");
        String nameFilter = WebUtils.getString(request, "nameFilter");
        // Pending docs
        List<Map<String,String>> pendingDocMaps = new ArrayList<Map<String,String>>();
        for ( Document doc : OKMDocument.getInstance().getChildren(null, "/okm:root") ) {
            Map<String,String> docMap = new HashMap<String,String>();
            docMap.put("name",doc.getPath().substring(doc.getPath().lastIndexOf("/") +1));
            docMap.put("uuid",doc.getUuid());
            docMap.put("path",doc.getPath());
            docMap.put("mimeType",doc.getMimeType());
            docMap.put("lastModified", df.format(doc.getLastModified().getTime()));
            pendingDocMaps.add(docMap);
        }
        ServletContext sc = getServletContext();
        sc.setAttribute("docPath",docPath);
        sc.setAttribute("nameFilter",nameFilter);
        sc.setAttribute("pendingDocuments",pendingDocMaps);
        sc.getRequestDispatcher("/sample/sample2/distributor.jsp").forward(request, response);
    }
}

```

distributor.jsp

The main layer.



Here JavaScript is used to control the height and width of the layer.

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
<html>
<head>

```

```

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="stylesheet" type="text/css" href="<%=request.getContextPath() %>/sample/
<link rel="stylesheet" type="text/css" href="<%=request.getContextPath() %>/css/jqu
<script type="text/javascript" src="<%=request.getContextPath() %>/js/jquery-1.7.1.
<script type="text/javascript" src="<%=request.getContextPath() %>/js/vanadium-min.
<script type="text/javascript" src="<%=request.getContextPath() %>/js/jquery.tables
</head>
<body>
<div id="leftPanel" style="float: left;">
  <jsp:include page="/sample/sample2/add_form.jsp"></jsp:include>
  <jsp:include page="/sample/sample2/full_list.jsp"></jsp:include>
</div>
<div id="centerPanel" style="float: left; width: 12px;">
  <div id="verticalSeparator" class="verticalSeparator"></div>
</div>
<div id="rightPanel" style="float: left;">
  <jsp:include page="/sample/sample2/pending.jsp"></jsp:include>
</div>

<script type="text/javascript">
  var width = $(window).width()-12; // Deleting middle panel
  var height = $(window).height();

  // Setting height
  $('#verticalSeparator').height(height);
  $('#leftPanel').height(height);
  $('#rightPanel').height(height);
  // Setting width
  $('#leftPanel').width(parseInt(width/2));
  $('#centerPanel').width(12);
  $('#rightPanel').width((width-parseInt((width/2))));
</script>
</body>
</html>

```

add_form.jsp

Form to upload a new document.

```

<%@ page import="com.openkm.frontend.client.constants.ui.UIFileUploadConstants" %>
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<h2 id="headerTitle_new_register">Upload new file</h2>
<form enctype="multipart/form-data" method="post" action="<%=request.getContextPath() %>/sample/sample2/action.jsp">
  <input type="hidden" name="path" value="/okm:root">
  <input type="hidden" name="action" value="<%=UIFileUploadConstants.ACTION_INSERT%>">
  <input type="hidden" name="message" value="">
  <input type="hidden" name="comment" value="">
  <input type="hidden" name="redirect" value="/sample/sample2/action.jsp">
  <table class="form">
    <tbody>
      <c:if test="${docPath != null && docPath != ''}">
        <tr>
          <td><div class="ok">Document sended</div></td>
        </tr>
      </c:if>
      <tr>
        <td>
          <div id="file_error" style="display:none; color : red;">File is mandatory.</div>
          <input type="File" name="file" class=":required;;file_error :only_on_blur">
        </td>
      </tr>
      <tr>
        <td align="center"><input type="submit" value="Send"></td>
      </tr>
    </tbody>
  </table>
</form>

```

```

    </tr>
  </tbody>
</table>
</form>

```

full_list.jsp

Show a list of documents.



Take a look at the ends where the JavaScript is used to control the scroll of the table.

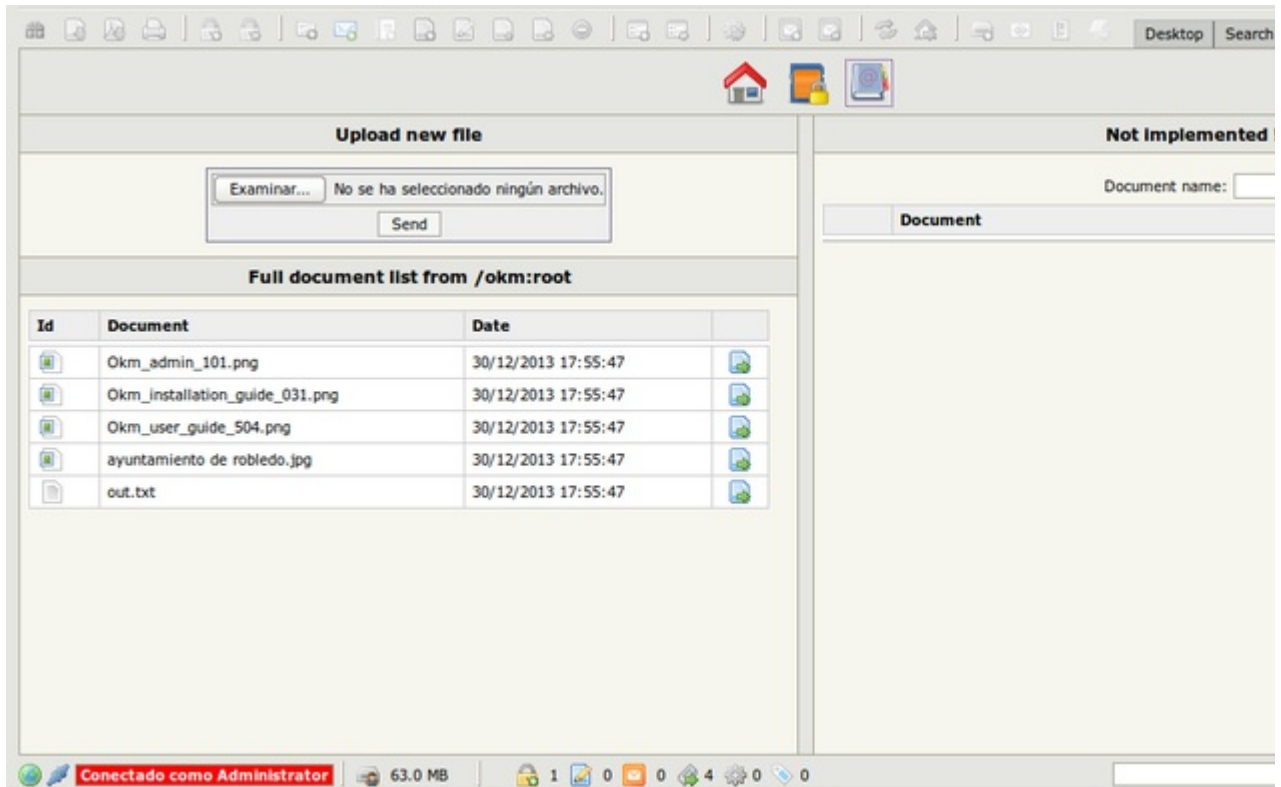
```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
<script type="text/javascript">
  // Note the parent.parent call because iframe is into other iframe
  function go(uuid) {
    alert('Document UUID: '+uuid);
    parent.parent.jsOpenPathByUuid(uuid);
  }
</script>
<h2 class="top">Full document list from /okm:root</h2>
<div align="center" style="padding: 0px 20px 0px 5px">
  <div class="tablescroll">
    <table id="docsPendingTable" width="100%" cellpadding="0" style="white-space: nowrap">
      <thead>
        <tr>
          <td>Id</td><td>Document</td><td>Date</td><td></td>
        </tr>
      </thead>
      <tbody>
        <c:set var="first" value="true"></c:set>
        <c:forEach var="document" items="${pendingDocuments}" varStatus="row">
          <c:choose>
            <c:when test="${first==true}">
              <tr class="first">
                <c:/when>
                <c:otherwise>
                  <tr>
                    <c:/otherwise>
                  </c:choose>
                  <td valign="center">
                  <td>${document.name}</td>
                  <td>${document.lastModified}</td>
                  <td align="center"><a href="javascript:go('${document.uuid}')">
                </tr>
                <c:set var="first" value="false"></c:set>
              </c:forEach>
            </tbody>
          </table>
        </div>
      </div>

      <script type="text/javascript">
        //Calculating table height removing from total other elements height
        var pendingTableHeight = $(window).height() - (parseInt($('#docsPendingTable').offsetTop));
        /**/
        jQuery(document).ready(function($)
        {
          $('#docsPendingTable').tableScroll({height:pendingTableHeight});
        });
      &lt;/script&gt;
</pre>
</div>
<div data-bbox="103 898 647 913" data-label="Page-Footer">
<p>Created in 2016 by OpenKM. OpenKM documentation is provided under <a href="#">OpenKM documentation license</a>.</p>
</div>
```

```
/*]]>*/  
</script>
```

Screenshot of sample2 view



Using iframe for extending frontend UI

This sample gives some clues about how to use iframe in frontend UI. This sample use ExtraTab feature with an iframe that contains references to OpenKM object as links.



The HTML file has a valid OpenKM UUID's links.

The sample below use [Exposed frontend JavaScript API](#).

File `extra_tab.html`:

```
<html>
<head>
  <title>Example</title>
</head>
<script type="text/javascript">
  function openPath(path) {
    parent.jsOpenPathByUuid(path);
  }
</script>
<body>
<h1>url</h1>
  <a href="javascript:void();" onclick="openPath('9789ac11-ac03-4ce7-87e9-9643b0ddf99."
</body>
</html>
```

Copy the `extra_tab.html` file into `$TOMCAT/webapps/OpenKM` folder.



Copy to OpenKM folder is mandatory, otherside from iframe will not be able to access protected resource.

When you change the contents of the `OpenKM.war` file is deployed a new version of the application and all the contents into `$TOMCAT_HOME/webapps/OpenKM` will be refreshed and you will lost any change you made directly in the `webapps/OpenKM` folder.

Finally configure ExtraTab feature with the new url, for example http://localhost:8080/OpenKM/extra_tab.html.



Consider http://localhost:8080/OpenKM/extra_tab.html as an example. You should use your own public OpenKM application url.

Exposed frontend JavaScript API



OpenKM UI is developed with GWT framework what helps on calling Java methods from JavaScript.
The UI frontend Java methods are public exposed as JavaScript functions and can be used in frontend context.








More information about GWT JSNI at [GWT Dev Codings Basics](#).

General functions



The methods that do not start with "js" will no longer be supported in the near future, we encourage you to move to js equivalent ones.

Function	Return	Description	Deprecated
jsI18n(String)	void	Retrieve a frontend translation.  The parameter is the translation key.	no
jsOpenPath(String, String)	void	Open some node in OpenKM UI.  First parameter is folder path. Second parameter is document path.  When document path parameter is empty will be opened a folder.	no
jsOpenPathByUuid(String)	void	Open some node in OpenKM UI.  The parameter is a valid node uuid (document, folder, mail or record).	no
jsRefreshFolder()	void	Refresh UI (like toolbar refresh button action).	no
jsWizard(String, String)	void	Execute wizard.  First parameter is a document path.	no



		<div>  Second parameter is a JSON to String serialization of GWTFileUploadResponse bean. </div> <div>  The wizard usually should be called after a uploading a document. </div>	
<i>i18n(String)</i>	<i>void</i>	Retrieve a frontend translation. <div>  The parameter is the translation key. </div>	yes
<i>openPath(String, String)</i>	<i>void</i>	Open some node in OpenKM UI. <div>  First parameter is folder path. Second parameter is document path. </div> <div>  When document path parameter is empty will be opened a folder. </div>	yes
<i>openPathByUuid(String)</i>	<i>void</i>	Open some node in OpenKM UI. <div>  The parameter is a valid node uuid (document, folder, mail or record). </div>	yes
<i>refreshFolder()</i>	<i>void</i>	Refresh UI (like toolbar refresh button action).	yes
<i>jsCancelCheckout()</i>	<i>void</i>	Cancel document checkout of the selected document in UI.	no
<i>jsGetActualPath()</i>	<i>String</i>	Return the path of selected tree node in navigator.	no
<i>jsGetActualNodePath()</i>	<i>String</i>	Return the path of the selected node in navigator or file brower.	no

HTML Editor functions

Function	Return	Description	Deprecated

<i>jsSearchDocumentHTMLEditorPopup()</i>	<i>void</i>	Show find document popup.	no
<i>jsSearchFolderHTMLEditorPopup()</i>	<i>void</i>	Show find folder popup.	no

Wiki functions

Function	Return	Description	Deprecated
<i>jsOpenWikiPage(String)</i>	<i>void</i>	Open a wiki page  The parameter is the unique wiki title.	no
<i>openWikiPage(String)</i>	<i>void</i>	Open a wiki page  The parameter is the unique wiki title.	yes

How to expose Java methods to JavaScript

Static method sample

Expose **static method** named ***i18n(String property)*** to JavaScript as "***jsI18n(String)***" function:

```
/**
 * initJavaScriptApi
 */
native void initJavaScriptApi() /*-{
  $wnd.jsI18n = function(s) {
    return @com.openkm.frontend.client.Main::i18n(Ljava/lang/String;) (s);
  };
}*/;
```

Non Static method sample

Expose **non static method** "***executeRefresh()***" from class `ToolBar.java` to JavaScript as `jsRefreshFolder` function:

In the `ToolBar` Java create a method named "***initJavaScriptApi()***":

```
public native void initJavaScriptApi(ToolBar toolBar) /*-{
  $wnd.jsRefreshFolder = function() {
    toolBar.@com.openkm.frontend.client.widget.toolbar.ToolBar::executeRefresh() ();
    return true;
  }
}*/;
```

After creating a new Toolbar object call the "**initJavaScriptApi()**" method:

```
ToolBar toolBar = new ToolBar();  
mainPanel.topPanel.toolBar.initJavaScriptApi(mainPanel.topPanel.toolBar);
```



Complete information about GWT JSNI at [GWT Dev Codings Basics](#).

Samples

Sample	Description
<i>Scripting samples</i>	<i>A collection of samples that use Scripting feature.</i>
<i>PHP SOAP Webservices samples</i>	<i>A collection of samples that use PHP and SOAP Webservices.</i>
<i>Crontab task - XML importer</i>	<i>A complete crontab task that imports files and set metadata groups using XML files.</i>

Scripting samples

Script	Description
<u>Script - Basic samples</u>	<ul style="list-style-type: none"> • Updates the repository statistics. • Shows the jBPM mail configuration. • Resets the user document size. • Refreshes the user items. • Creates folders and sets property group. • Shows the number of documents, folders and size from a given path. • Gets path by UUID. • Shows all documents used in every running workflow.
<u>Script - Cognitive PDF/A</u>	Makes PDF/A ISO standard documents from scanned document images.
<u>Script - Compact documents with size equals to 0</u>	Finds all documents with size=0 and compact history of versions to latest.
<u>Script - Count all children subfolder</u>	Counts all children folders - and sub folders to, recursion - into first level folder (okm:root children folders).
<u>Script - Exporter</u>	The repository exports execution based on crontab task.
<u>Script - Export repository folder to server as a ZIP file</u>	Exports the folder contents to the server application as a ZIP file.
<u>Script - Folders deep</u>	Finds the folders with some deep.
<u>Script - Generate database script</u>	Generates the database script set to hibernate.cfg.xml file.
<u>Script - Import data from CSV file</u>	Imports metadata group values from csv file.
<u>Script - Import reports from a folder</u>	Imports reports from \$TOMCAT_HOME/reports also enables them in the default profile.

<u>Script - Purge all folders into some parent</u>	<i>Purges all folders into some parent, without previously moving files to trash.</i>
<u>Script - Purge all users trashes</u>	<i>Example of how to clean all users trashes.</i>
<u>Script - Recursive character renaming</u>	<i>Renames a character in all folders or documents from some initial path.</i>
<u>Script - Recursive repository traversal</u>	<i>Navigates across all repository nodes recursively.</i>
<u>Script - Recursive change security</u>	<i>Change security.</i>
<u>Script - Security change at low level</u>	<i>Changes the security at database level.</i>
<u>Script - Correct security in personal nodes</u>	<i>Correct security in personal nodes.</i>

Script - Basic samples

Update repository statistics

Refresh the [Statistics](#).

```
import com.openkm.core.*;

new RepositoryInfo().run();
```

Show jBPM mail configuration

```
import org.jbpm.*;

print("Templates: " + JbpmConfiguration.Configs.getString("resource.mail.templates"));
print("From: " + JbpmConfiguration.Configs.getString("jbpm.mail.from.address") + "<br>");
print("Host: " + JbpmConfiguration.Configs.getString("jbpm.mail.smtp.host") + "<br/>")
```

Reset user document size



The application assigns a specific size from the repository to some user (it's used by user quota exceed checking).

```
import com.openkm.dao.bean.cache.*;
import com.openkm.cache.*;

UserItems ui = UserItemsManager.get("okmAdmin");
ui.setSize(0);
```

Refresh user items



The application internally assign documents to users (also used by user quota exceed checking)

```
import com.openkm.cache.*;

UserItemsManager.refreshDbUserItems();
```

Create some folders and set property group

```
import com.openkm.api.*;

for (int i=0; i < 10; i++) {
    String path = "/okm:root/fld_" + i;
    OKMFolder.getInstance().createSimple(null, path);
    OKMPropertyGroup.getInstance().addGroup(null, path, "okg:technology");
}
```

Show number of documents, folders and size from a given path

```
import com.openkm.bean.*;
import com.openkm.util.*;
import com.openkm.api.*;

ContentInfo ci = OKMFolder.getInstance().getContentInfo(null, "/okm:root/path/to/folder");
print("Folders: " + ci.getFolders() + "<br/>");
print("Documents: " + ci.getDocuments() + "<br/>");
print("Size: " + FormatUtil.formatSize(ci.getSize()) + "<br/>");
```

Get path by UUID

```
import com.openkm.api.OKMRepository;
String path = OKMRepository.getInstance().getNodePath(null, "b99f7973-4b90-457a-a2d0-000000000000");
print(path);
```

Show all documents used in every running workflow

```
import com.openkm.module.common.CommonWorkflowModule;
import com.openkm.bean.workflow.ProcessDefinition;
import com.openkm.bean.workflow.ProcessInstance;
import com.openkm.core.Config;

for (ProcessDefinition procDef : CommonWorkflowModule.findAllProcessDefinitions()) {
    for (ProcessInstance procIns : CommonWorkflowModule.findProcessInstances(procDef)) {
        if (procIns.getEnd() == null) {
            String uuid = (String) procIns.getVariables().get(Config.WORKFLOW_PROCESS_UUID);
            print("Doc UUID: " + uuid + ", Process Definition: " + procDef.getId() + "\n");
        }
    }
}
```

Script - Cognitive PDF/A

[Cognitive PDF/A](#) is a powerful tool [Cognitive Forms](#) which it makes PDF/A ISO standard documents from scanned document images. OpenKM integrate Compressimo transparently using automation or crontab utilities.



- Input server folder where images are stored at `systemFolder = "c:\\samples"`
- The `cognitivePdf` variable is the Cognitive PDF/A server path.
- The OpenKM path where compressimo files are imported `String path = "/okm:root/imported" + "/" + uploadFile.getName()`

```
import java.io.*;
import java.util.*;
import com.openkm.module.db.stuff.DbSessionManager;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.openkm.bean.Document;
import com.openkm.api.OKMDocument;

String token = DbSessionManager.getInstance().getSystemToken();
String systemFolder = "c:\\samples";
String cognitivePdfa = "C:\\Users\\openkm\\Desktop\\cpdfa-1.4.4-dev-win32-1month\\cpd
Logger log = LoggerFactory.getLogger("com.openkm.compressimo");

// Loading files
log.info("***** Process BEGIN *****");
File folder = new File(systemFolder);
File[] listOfFiles = folder.listFiles();

for (int i=0; i<listOfFiles.length; i++) {
    File file = listOfFiles[i];

    if (file.isFile() && file.getName().toLowerCase().endsWith(".tif")) {
        // Converting tif file to pdf
        String tifFile = systemFolder + "\\\" + file.getName();
        String pdfFile = systemFolder + "\\\" + file.getName().toLowerCase().replace(
        log.info("Converting file:" + tifFile+" to:"+pdfFile);
        String command = cognitivePdfa + tifFile + " " + pdfFile;
        Process process = Runtime.getRuntime().exec(command);

        BufferedReader input = new BufferedReader(new InputStreamReader(process.getIn
        String line = null;

        while ((line=input.readLine()) != null) {
            log.info(line);
        }

        int exitVal = process.waitFor();
        log.info("Exit code " + exitVal);

        // Upload PDF file
        File uploadFile = new File(pdfFile);
        Document doc = new Document();
        String path = "/okm:root/imported" + "/" + uploadFile.getName();
        log.info("Importing file:"+ pdfFile + " to:"+path);
        doc.setPath(path);
        FileInputStream fis = new FileInputStream(uploadFile);
```

```
        doc = OKMDocument.getInstance().create(token, doc, fis);
        fis.close();

        // Deleting pdf File
        uploadFile.delete();
    }

    log.info("***** Process END *****");
```

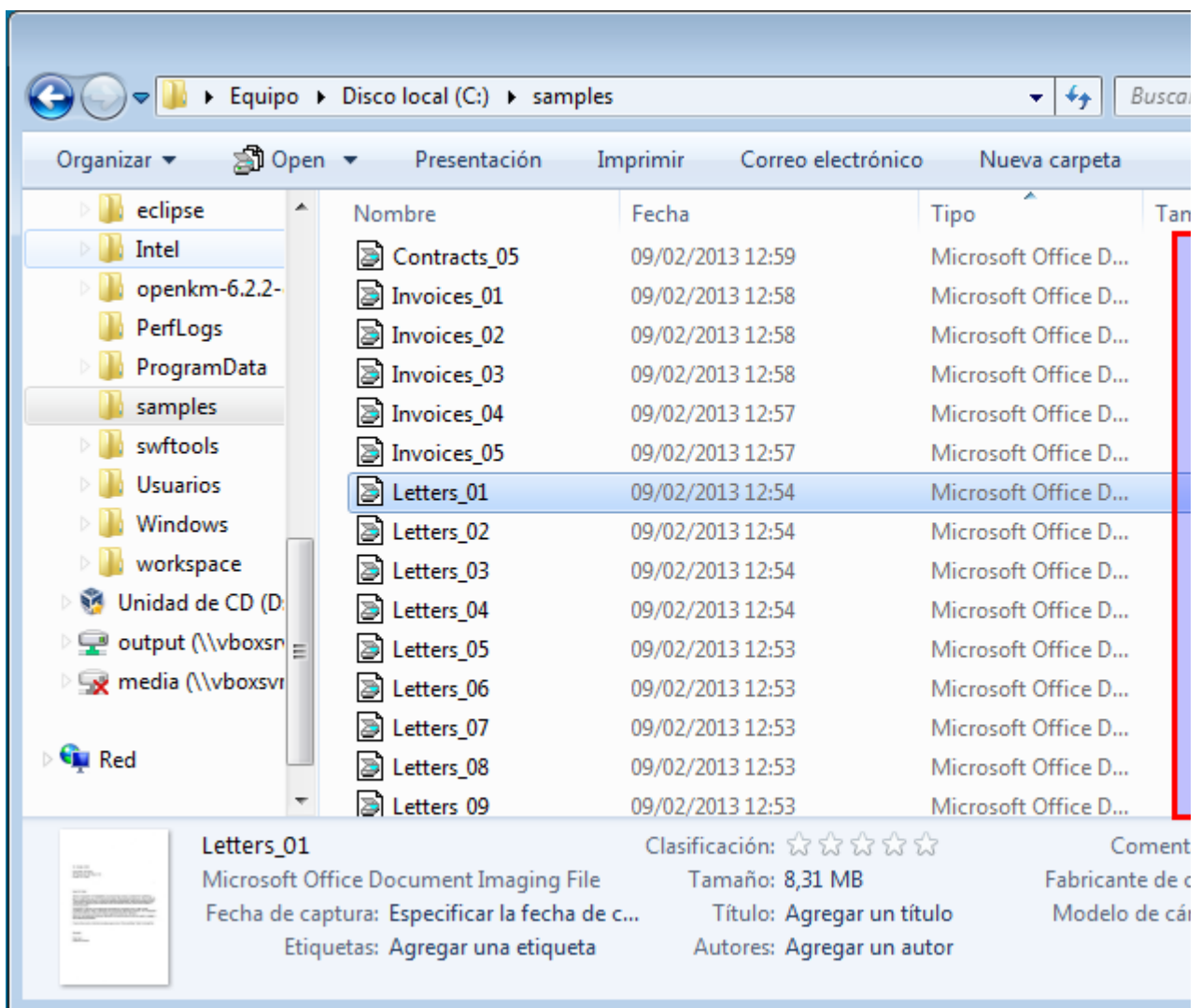
For more information:

- [Cognitive PDF/A documentation](#).
- [Cognitive PDF/A website](#).
- [Cognitive Forms](#).

Images

Scanned files

Red square remark initial document size.



Crontab compressimo Importer task

Crontab					
Name	Expression	Mime	File Name	Mail	Last B
User Items Serialize	@hourly	application/x-bsh	UserItemsSerialize.bsh	noreply@openkm.com	Feb 10, 2 12:00:00
User Mail Importer	*/30 * * *	application/x-bsh	UserMailImporter.bsh	noreply@openkm.com	Feb 10, 2 12:00:00
Text Extractor Worker	*/5 * * * *	application/x-bsh	TextExtractorWorker.bsh	noreply@openkm.com	Feb 10, 2 12:00:00
Repository Info	@daily	application/x-bsh	RepositoryInfo.bsh	noreply@openkm.com	
Session Watchdog	*/5 * * * *	application/x-bsh	SessionWatchdog.bsh	noreply@openkm.com	Feb 10, 2 12:00:00
Compressimo importer	0 0 0 0 0	application/x-bsh	compressimo.bsh		Feb 10, 2 11:35:03

OpenKM log server output of the importer task execution

```

2013-02-10 11:35:03,724 [http-bio-0.0.0.0-8080-exec-6] INFO com.openkm.compressino - ***** P
2013-02-10 11:35:03,749 [http-bio-0.0.0.0-8080-exec-6] INFO com.openkm.compressino - Convert
2013-02-10 11:35:10,399 [http-bio-0.0.0.0-8080-exec-6] INFO com.openkm.compressino - 1 page(
2013-02-10 11:35:10,404 [http-bio-0.0.0.0-8080-exec-6] INFO com.openkm.compressino - Result
2013-02-10 11:35:11,691 [http-bio-0.0.0.0-8080-exec-6] INFO com.openkm.compressino - Exit co
2013-02-10 11:35:11,694 [http-bio-0.0.0.0-8080-exec-6] INFO com.openkm.compressino - Importi
2013-02-10 11:35:11,752 [http-bio-0.0.0.0-8080-exec-6] INFO com.openkm.vernun.VersionNumerat
erationAdapter
2013-02-10 11:35:11,934 [http-bio-0.0.0.0-8080-exec-6] INFO com.openkm.compressino - Convert
2013-02-10 11:35:18,382 [http-bio-0.0.0.0-8080-exec-6] INFO com.openkm.compressino - 1 page(
2013-02-10 11:35:18,390 [http-bio-0.0.0.0-8080-exec-6] INFO com.openkm.compressino - Result
2013-02-10 11:35:19,682 [http-bio-0.0.0.0-8080-exec-6] INFO com.openkm.compressino - Exit co
2013-02-10 11:35:19,687 [http-bio-0.0.0.0-8080-exec-6] INFO com.openkm.compressino - Importi

```

Imported files into OpenKM

	Name	Size	Update date	Author	Version
	banking_01.pdf	227.6 KB	10-02-2013 11:35:11	system	1.0
	banking_02.pdf	191.9 KB	10-02-2013 11:35:19	system	1.0
	banking_04.pdf	315.9 KB	10-02-2013 11:35:26	system	1.0
	banking_05.pdf	206.4 KB	10-02-2013 11:35:33	system	1.0
	invoices_01.pdf	255.7 KB	10-02-2013 11:35:39	system	1.0
	invoices_02.pdf	196.0 KB	10-02-2013 11:35:46	system	1.0
	invoices_03.pdf	122.0 KB	10-02-2013 11:35:51	system	1.0
	invoices_04.pdf	119.7 KB	10-02-2013 11:35:56	system	1.0
	invoices_05.pdf	91.9 KB	10-02-2013 11:36:02	system	1.0
	letters_01.pdf	141.2 KB	10-02-2013 11:36:07	system	1.0
	letters_02.pdf	162.6 KB	10-02-2013 11:36:13	system	1.0
	letters_03.pdf	175.0 KB	10-02-2013 11:36:19	system	1.0
	letters_04.pdf	141.3 KB	10-02-2013 11:36:25	system	1.0
	letters_05.pdf		10-02-2013 11:36:28	system	1.0

Properties Notes Security History Preview

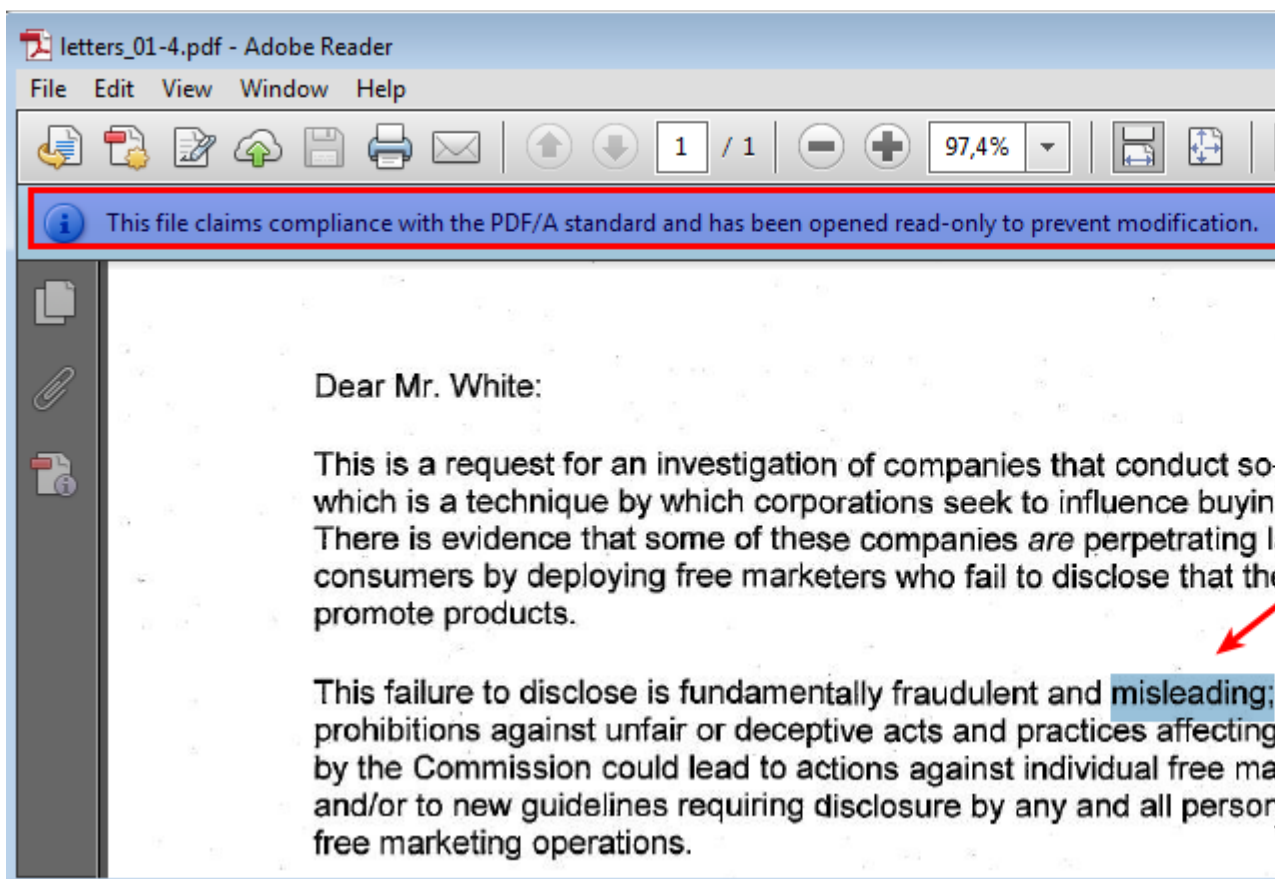
115%

1 / 1

Dear Mr. White:

This is a request for an investigation of companies that conduct so-called "free marketing," which is a technique by which corporations seek to influence buying decisions, often by stealth. There is evidence that some of these companies are perpetrating large-scale deception upon consumers by deploying free marketers who fail to disclose that they have been enlisted to promote products.

PDF/A compliance



Search engine results by content

Basic Advanced Metadata

Context Taxonomy

Content misleading

Name

Keywords

User

Date range between

View advanced mode

Results compact view

View property groups

Save as user news

Save search

Page results 10

Clean Search

Results : 1 1 to 1

letters_01.pdf 1.0

to disclose is fundamentally fraudulent and misleading; and it might violate prohibitions against unfair or deceptive acts and practices affecting commerce. An investigation by the Commission could lead to actions against individual free marketers, and/or to new ...

Folder: /okm:root/imported

Author: system Size: 141.2 KB Version: 1.0 Update date: 10-02-2013 11:36:07

Script - Compact documents with size equals to 0

Find all documents with size=0 and compact history of versions to latest.

```
import java.util.*;
import com.openkm.api.OKMDocument;
import com.openkm.api.OKMRepository;
import com.openkm.dao.HibernateUtil;
import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.Query;
import com.openkm.dao.bean.NodeDocumentVersion;

String qs = "from NodeDocumentVersion ndv where ndv.size=0";
Session session = HibernateUtil.getSessionFactory().openSession();
Transaction tx = session.beginTransaction();
Query q = session.createQuery(qs);
List docVersionList = q.list();
HibernateUtil.commit(tx);

print("Number of nodes: "+docVersionList.size()+"<br/>");

for (NodeDocumentVersion ndv : docVersionList) {
    String path = OKMRepository.getInstance().getNodePath(null, ndv.getParent());
    OKMDocument.getInstance().purgeVersionHistory(null, path);
}

HibernateUtil.close(session);
print("done");
```


Script - Count all children subfolder

Count all children folders - and subfolders into, recursion - into first level folder (okm:root children folders).

```
import com.openkm.api.*;
import com.openkm.bean.*;
import org.hibernate.Session;
import org.hibernate.Transaction;
import com.openkm.dao.HibernateUtil;
import org.hibernate.HibernateException;
import org.hibernate.Query;

class Util {
    public int childCount(String uuid) {
        String qsUUIDList = "select uuid from NodeFolder nf where nf.parent=:parent";
        Session session = null;
        Transaction tx = null;
        int value = 0;
        List uuidList = new ArrayList();

        try {
            session = HibernateUtil.getSessionFactory().openSession();
            tx = session.beginTransaction();
            Query q = session.createQuery(qsUUIDList).setCacheable(true);
            q.setString("parent", uuid);
            uuidList = q.list();
            HibernateUtil.commit(tx);
            value = uuidList.size();
            print(value);
        } catch (HibernateException e) {
            HibernateUtil.rollback(tx);
        } finally {
            HibernateUtil.close(session);
        }

        for (String childUuid : uuidList) {
            value = value + childCount(childUuid);
        }
        return value;
    }
}

Util util = new Util();
for (Folder fld : OKMFolder.getInstance().getChildren(null, "/okm:root")) {
    print(fld.getPath()+"->"+util.childCount(fld.getUuid())+"<br/>");
}
```

Script - Exporter

Repository export execution based on crontab task.

```
import com.openkm.util.FileLogger;
import com.openkm.bean.ContentInfo;
import com.openkm.api.*;
import com.openkm.util.impexp.*;
import java.io.*;
import java.text.*;
import com.openkm.core.Config;
import java.util.*;
import com.openkm.module.db.stuff.*;

// Configuration parameters
String token = DbSessionManager.getInstance().getSystemToken();
String LOG_FILE_NAME = "CrontabRepositoryExporter";
String repoPath = "/okm:root";
String fsPath = "F:\\Restore_OKM";
boolean metadata = false;
boolean history = false;

// Start
FileWriter fw = null;
try {
    FileLogger.info(LOG_FILE_NAME, "Started");
    // Open outputstream
    String fileDate = new SimpleDateFormat("yyyyMMdd").format(new Date());
    String fileName = Config.LOG_DIR + File.separator + LOG_FILE_NAME + "_" + fileDate;
    File file = new File(fileName);
    fw = new FileWriter(file);
    // Starting the process
    ContentInfo cInfo = OKMFolder.getInstance().getContentInfo(token, repoPath);
    FileLogger.info(LOG_FILE_NAME, "Files & directories to export:"+(cInfo.getDocumentCount()));
    File dir = new File(fsPath);
    ImpExpStats stats = RepositoryExporter.exportDocuments(token, repoPath, dir, metadata, history);
    FileLogger.info("CrontabRepositoryExporter", "Finished");
} catch (Exception e) {
    try {
        if (fw != null) {
            fw.close();
        }
    } catch (IOException e1) {
        // Ignore
    }
}
```

Script - Export repository folder to server as a ZIP file

This script exports all contents of an OpenKM folder to application server as a ZIP file.

```
import java.io.File;
import java.io.FileOutputStream;
import java.io.StringWriter;
import com.openkm.util.FileUtils;
import com.openkm.util.impexp.RepositoryExporter;
import com.openkm.util.impexp.TextInfoDecorator;
import com.openkm.util.ArchiveUtils;
import org.apache.commons.io.IOUtils;

// Destination
File file = new File("/home/openkm/import/folderExport.zip");
// OpenKM folder contents to be exported
String fldPath = "/okm:root/invoices";

FileOutputStream os = new FileOutputStream(file);
StringWriter out = new StringWriter();
File tmp = FileUtils.createTempDir();
RepositoryExporter.exportDocuments(null, fldPath, tmp, false, false, out, new TextInfoDecorator());
ArchiveUtils.createZip(tmp, "import", os);
org.apache.commons.io.FileUtils.deleteDirectory(tmp);
IOUtils.closeQuietly(out);
os.flush();
os.close();
```

Script - Folders deep

The script finds all folders with some deep.



The variable **MAX_DEPTH=5** indicates the folder deep.

```
import org.hibernate.HibernateException;
import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.Transaction;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.openkm.core.DatabaseException;
import com.openkm.dao.bean.NodeBase;
import com.openkm.dao.HibernateUtil;
import com.openkm.core.Config;
import com.openkm.api.OKMRepository;
import com.openkm.api.OKMFolder;

Logger log = LoggerFactory.getLogger("com.openkm.folder.deep");
int MAX_DEPTH = 5;
int count = 0;

void findNodePathHelper(Session session, NodeBase parentNode, int depth) throws Hiber
    String qs = "from NodeBase nb where nb.parent=:parent";
    Query q = session.createQuery(qs).setCacheable(true);
    q.setString("parent", parentNode.getUuid());

    for (NodeBase nb : q.list()) {
        if (depth == MAX_DEPTH && OKMFolder.getInstance().isValid(null, OKMRepository
            print("uuid:" + nb.getUuid() + " path:" + OKMRepository.getInstance().getNod
        ) else if (depth < MAX_DEPTH) {
            findNodePathHelper(session, nb, depth + 1);
        }
    }
}

log.info("***** Process BEGIN *****");
String qs = "from NodeBase nb where nb.parent=:parent";
Session session = null;

try {
    session = HibernateUtil.getSessionFactory().openSession();
    session.beginTransaction();

    // First level nodes
    Query q = session.createQuery(qs);
    q.setString("parent", Config.ROOT_NODE_UUID);

    for (NodeBase nb : q.list()) {
        // Process in depth
        findNodePathHelper(session, nb, 0);
    }

    HibernateUtil.commit(session.getTransaction());
} catch (HibernateException e) {
    HibernateUtil.rollback(session.getTransaction());
    throw new DatabaseException(e.getMessage(), e);
} finally {
    HibernateUtil.close(session);
}
```

```

}

log.info("***** Process END *****");

```

Images

Execution

```

1 // com.openkm.dao.NodeBaseDAO.getInstance().fixNodePath();
2 import org.hibernate.HibernateException;
3 import org.hibernate.Query;
4 import org.hibernate.Session;
5 import org.hibernate.Transaction;
6 import org.slf4j.Logger;
7 import org.slf4j.LoggerFactory;
8 import com.openkm.core.DatabaseException;
9 import com.openkm.dao.bean.NodeBase;
10 import com.openkm.dao.HibernateUtil;
11 import com.openkm.core.Config;
12 import com.openkm.api.OKMRepository;
13 import com.openkm.api.OKMFolder;
14
15 Logger log = LoggerFactory.getLogger("com.openkm.folder.deep");
16 int MAX_DEPTH = 5;
17 int count = 0;
18
19 void findNodePathHelper(Session session, NodeBase parentNode, int depth) throws HibernateException {
20     String qs = "from NodeBase nb where nb.parent=:parent";
21     Query q = session.createQuery(qs).setCacheable(true);
22     q.setString("parent", parentNode.getUuid());
23
24     for (NodeBase nb : q.list()) {
25         if (depth == MAX_DEPTH && OKMFolder.getInstance().isValid(null, OKMRepository.getInstance().getNodePath(null, nb.getUuid(), nb.getUuid())) {
26             print("uuid:" + nb.getUuid() + " path:" + OKMRepository.getInstance().getNodePath(null, nb.getUuid(), nb.getUuid()) + "<br/>");
27         } else if (depth < MAX_DEPTH) {
28             findNodePathHelper(session, nb, depth + 1);
29         }
30     }
31 }
32
33 log.info("***** Process BEGIN *****");
34 String qs = "from NodeBase nb where nb.parent=:parent";
35 Session session = null;
36
37 try {
38     session = HibernateUtil.getSessionFactory().openSession();
39     session.beginTransaction();
40
41     // First level nodes
42     Query q = session.createQuery(qs);
43     q.setString("parent", Config.ROOT_NODE_UUID);
44
45     for (NodeBase nb : q.list()) {
46         // Process in depth
47         findNodePathHelper(session, nb, 0);
48     }
49 }

```

Results

Script error
Script result
Script output
<pre> uuid:9eb8371e-c105-4cfa-9cb7-6b208f0ec30d path:/okm:root/test/test2/test3/test4/test5 uuid:c9567359-c1a6-4edb-a82b-0c72dbfd67b1 path:/okm:root/test1/test2/test3/test4/test5 uuid:8c4ef27c-f851-46d7-8b5d-e5bdeff12907 path:/okm:trash/okmAdmin/pacientes/paciente1/2012/facturas uuid:c24d3863-6df8-4b29-8f57-3f29d5d33196 path:/okm:trash/okmAdmin/tres/cuatro2/sis/manolo </pre>

Script - Generate database script

Generates the database schema present at `hibernate.cfg.xml` file.



The `hibernate.cfg.xml` schema file only contains jBPM database structure.

You can find the DDL SQL sentences for table creation for several databases at [Jbpm-ddl-3.3.1.zip](#).

```
import java.io.File;

import org.hibernate.cfg.Configuration;
import org.hibernate.tool.hbm2ddl.SchemaExport;

Configuration cfg = new Configuration();
cfg.configure(new File("/desarrollo/tomcat/tomcat-7.0.27-openkm-6.3/webapps/OpenKM/WEB-INF/classes/hibernate.cfg.xml"));
SchemaExport se = new SchemaExport(cfg);
se.setOutputFile("/home/openkm/schema.sql");
se.setFormat(true);
se.create(false, false);
```

See also:

- [How To Generate DDL Scripts from Hibernate](#)

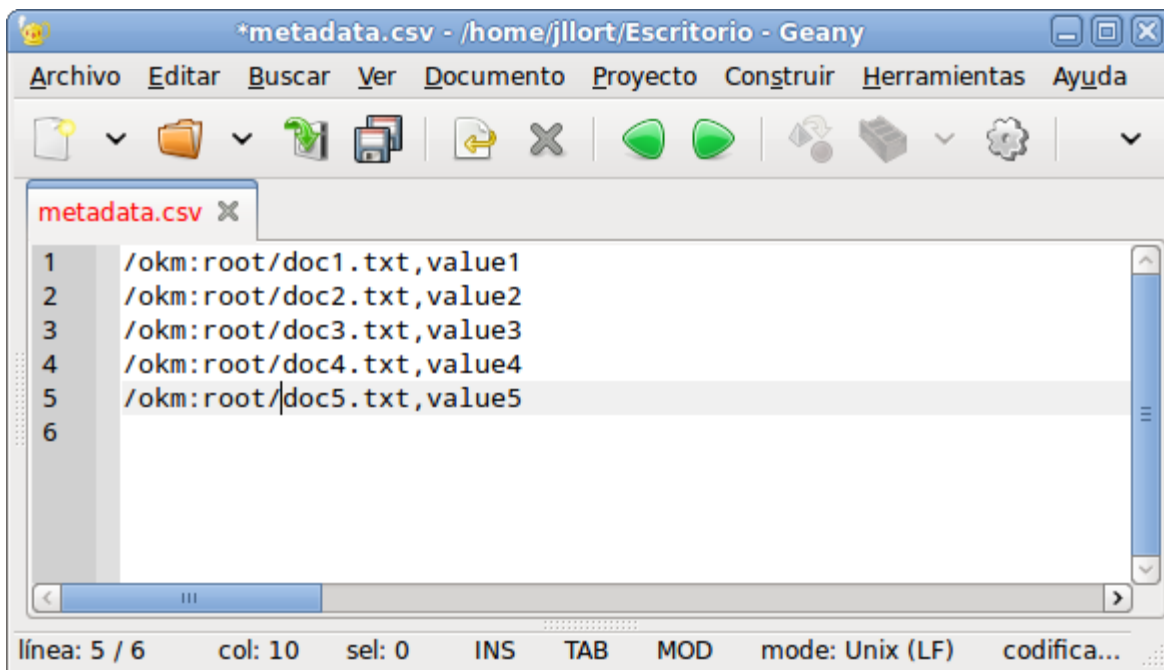
Script - Import data from CSV file

The script can be executed from the administration scripting (or used as crontab script too) to import metadata values.



- There's a metadata CSV file with two columns, one for the document path and other for the metadata value. The script can be easily changed to use more than two columns.
- Script searches documents by document name. In theory there's only one document with the same name in the system, otherwise script shows error. Here could be used document path from first csv column, but to get more complete sample it has been used a search engine to find document.
- For each document found, script add metadata group with CSV values.

The CSV file has two column, first column contains document path and second column has metadata value.



Download CSV sample file [csv_metadata.zip](#).

The metadata group definition:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.1//EN"
"http://www.openkm.com/dtd/property-groups-2.1.dtd">
<property-groups>
  <property-group label="Metadata" name="okg:metadata">
    <input label="Document ID" name="okp:metadata.value" type="text"/>
  </property-group>
</property-groups>
```

The script:

```
import java.io.FileReader;
import java.io.Reader;
import java.util.Iterator;
import java.util.List;
import java.util.Collection;

import com.googlecode.jcsv.CSVStrategy;
import com.googlecode.jcsv.reader.CSVReader;
import com.googlecode.jcsv.reader.internal.CSVReaderBuilder;
import com.googlecode.jcsv.reader.internal.DefaultCSVEntryParser;

import com.openkm.dao.bean.QueryParams;
import com.openkm.bean.QueryResult;
import com.openkm.api.OKMSearch;
import com.openkm.util.FileLogger;
import com.openkm.api.OKMPropertyGroup;
import com.openkm.util.PathUtils;
import com.openkm.bean.Document;

String grpName = "okg:metadata";
String FILE_LOG_NAME = "CSVLOG";
String META_PATH = "/home/openkm/csv/";
String META_FILE_NAME = "metadata.csv";
int uniqueFileName = 0;
int valueColumn = 1;

// Format definition
char delimiter = ',';
char quoteCharacter = '"';
char commentIndicator = '#';
boolean skipHeader = true;
boolean ignoreEmptyLines = true;
CSVStrategy strategy = new CSVStrategy(delimiter, quoteCharacter, commentIndicator, skipHeader, ignoreEmptyLines);
// File reader
Reader reader = new FileReader(META_PATH + META_FILE_NAME);
// CSV reader
CSVReader csvParser = new CSVReaderBuilder(reader).strategy(strategy).entryParser(new DefaultCSVEntryParser());
List data = csvParser.readAll();
int count = 1;
int countFound = 0;
int countNotDocument = 0;
int moreThanOneDocumentFound = 0;
int notFound = 0;
int noName = 0;

for (Iterator it = data.listIterator(); it.hasNext();) {
    String[] row = (String[]) it.next();
    String docPath = row[valueColumn];
    print(count + ">>>> " + docPath);

    if (docPath != null && !docPath.equals("")) {
        QueryParams queryParams = new QueryParams();
        queryParams.setDomain(QueryParams.DOCUMENT);
        queryParams.setName(PathUtils.getName(docPath));
        Collection results = OKMSearch.getInstance().find(null, queryParams);

        if (results.size() == 1) {
            QueryResult queryResult = (QueryResult) results.iterator().next();
            if (queryResult.getDocument() != null) {
                print("found");
                countFound++;
                // Add Group
                OKMPropertyGroup.getInstance().addGroup(null, docPath, grpName);
            }
        }
    }
}
```



```

        // Add metadata
        Map map = new HashMap();
        map.put("okp:metadata.value", row[valueColumn]);
        OKMPropertyGroup.getInstance().setPropertiesSimple(null, docPath, grp);
    } else {
        print("error is not document");
        countNotDocument++;
    }
} else if (results.size() > 1) {
    print("error more than one document found can not decide");
    moreThanOneDocumentFound++;
} else {
    print("not found");
    notFound++;
}
} else {
    print("error document has no name");
    noName++;
}

print("</br>");

//FileLogger.info(FILE_LOG_NAME, "Document name ''{0}'' to ''{1}''", row[0], row[1],
count++);
}

print("Total:" + count + "</br>");
print("Found:" + countFound + "</br>");
print("Error not document:" + countNotDocument + "</br>");
print("Error more then one document found:" + moreThanOneDocumentFound + "</br>");
print("Error not found:" + notFound + "</br>");
print("Error name empty:" + notFound + "</br>");

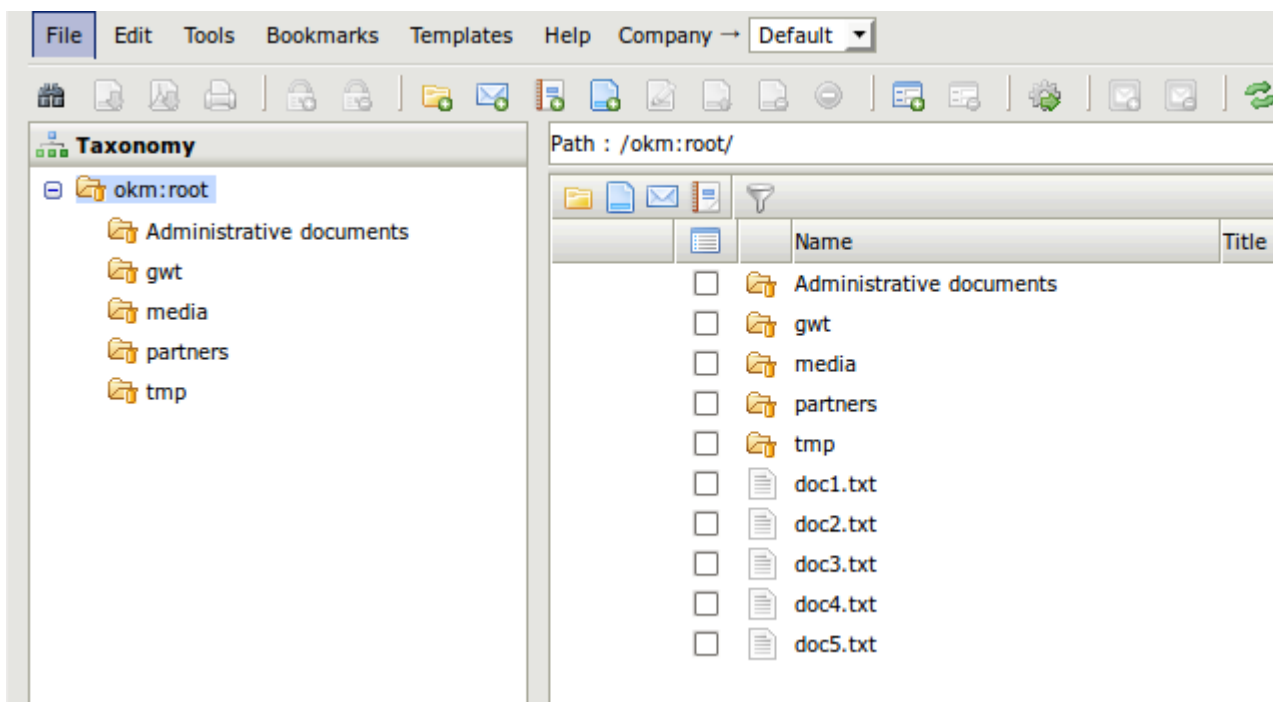
```

Image

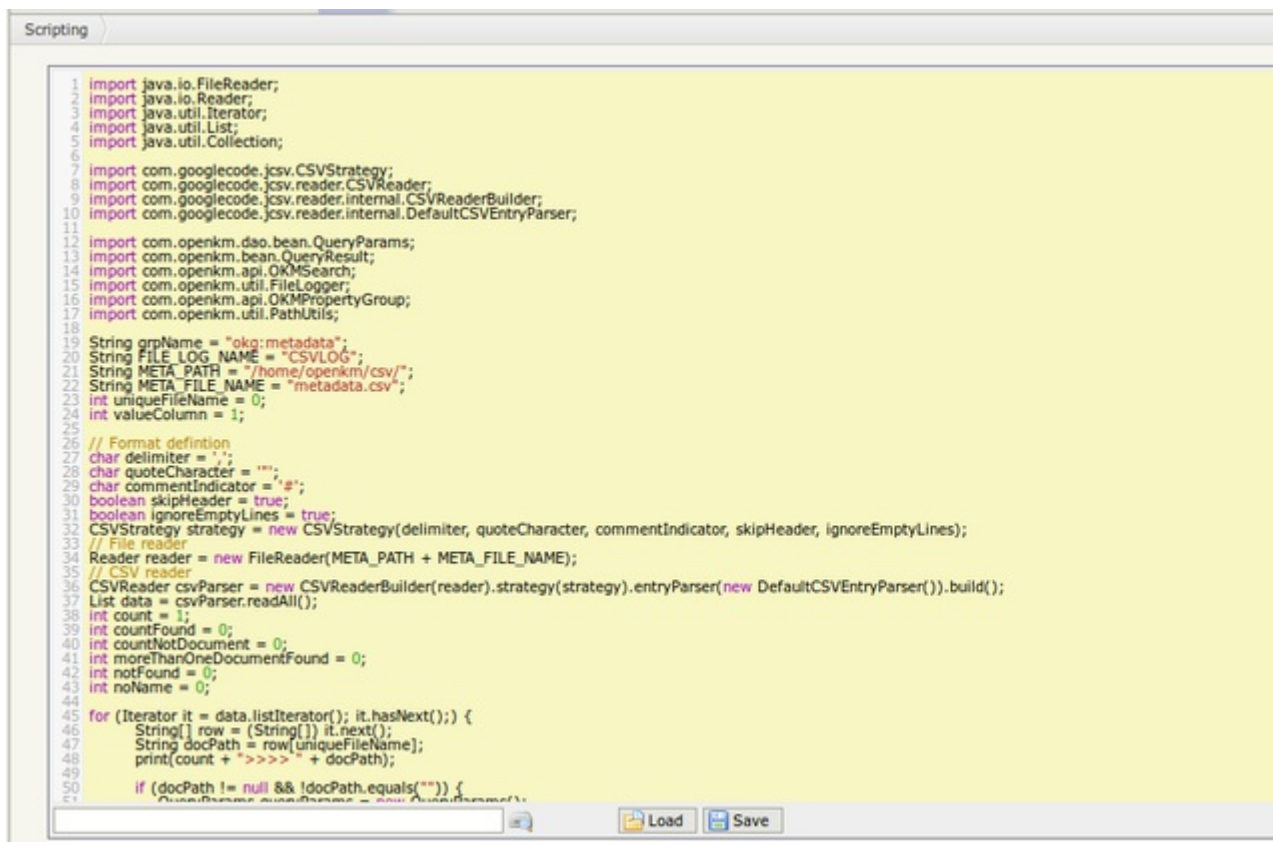
Register the metadata group

Group label		Group name			
Metadata		okg:metadata			
Label	Name	Width	Height	Field	
Document ID	okp:metadata.value	100px	25px	Input	Readonly: false Data: Type: text

Repository view



Execution



Script results

Time: 00:00:00.374

Script error
Script result
Script output
1>>>> /okm:root/doc2.txt found 2>>>> /okm:root/doc3.txt found 3>>>> /okm:root/doc4.txt found 4>>>> /okm:root/doc5.txt found Total:5 Found:4 Error not document:0 Error more then one document found:0 Error not found:0 Error name empty:0

Imported metadata

Path : /okm:root/

	Name	Title	Language	Size	U
<input type="checkbox"/>	Administrative documents				1
<input type="checkbox"/>	gwt				1
<input type="checkbox"/>	media				1
<input type="checkbox"/>	partners				0
<input type="checkbox"/>	tmp				2
<input type="checkbox"/>	doc1.txt			2.2 KB	3
<input type="checkbox"/>	doc2.txt			2.2 KB	3
<input type="checkbox"/>	doc3.txt			2.2 KB	3
<input type="checkbox"/>	doc4.txt			2.2 KB	3
<input type="checkbox"/>	doc5.txt			2.2 KB	3

Properties **Metadata** Change Delete

Document ID value2

Script - Import reports from a folder

Import reports from \$TOMCAT_HOME/reports and also enable them in the default profile.

```
import java.io.*;
import java.sql.*;
import org.hibernate.*;
import com.openkm.core.*;
import com.openkm.util.*;
import com.openkm.dao.*;

Session hbmSession = HibernateUtil.getSessionFactory().openSession();
Connection con = hbmSession.connection();
Statement st = con.createStatement();
File reports = new File(Config.HOME_DIR + "/reports");

try {
    if (reports.isDirectory()) {
        for (File rep : reports.listFiles()) {
            int id = ReportDAO.createFromFile(rep, FileUtils.getFileName(rep.getName()));
            String sql = "insert into OKM_PROFILE_MSC_REPORT (PRP_ID, PRP_REPORT) values (" + id + ", '" + rep.getName() + "')";
            LegacyDAO.execute(con, sql);
        }
    }
} catch (Exception e) {
    e.printStackTrace();
} finally {
    LegacyDAO.close(con);
    HibernateUtil.close(hbmSession);
}
```

Script - Purge all folders into some parent

That script purges all folders into a parent, without previously moving files to trash.



The purge action is a non recovery action that removes definitely the objects from OpenKM repository. Use it with care.

```
import org.hibernate.*;
import com.openkm.dao.*;
import com.openkm.dao.bean.NodeFolder;
import java.util.*;
import com.openkm.api.OKMFolder;

Session session = null;
found = true;

try {
    String sql = "from NodeFolder nf where nf.parent='aa08b392-0117-4b29-af9a-c4419ba";
    while (found) {
        session = HibernateUtil.getSessionFactory().openSession();
        Query q = session.createQuery(sql);
        q.setMaxResults(100);
        found = false;
        List fldList = new ArrayList();

        for (NodeFolder fld : q.list()) {
            found = true;
            fldList.add(fld.getUuid());
        }

        for (String uuid : fldList) {
            try {
                OKMFolder.getInstance().purge(null, uuid);
            } catch (Exception e) {
                print(e.getMessage() + "<br/>");
            }
        }
    }
} catch (Exception e) {
    print(e.getMessage());
} finally {
    HibernateUtil.close(session);
}
```

Script - Purge all users trashes

```
import com.openkm.api.*;
import com.openkm.core.*;
import com.openkm.bean.*;
import com.openkm.module.db.stuff.*;

String token = DbSessionManager.getInstance().getSystemToken();

for (Folder trash : OKMFolder.getInstance().getChildren(token, "/okm:trash")) {
    print("Trash: " + trash.getPath() + "<br/>");

    for (Folder fld : OKMFolder.getInstance().getChildren(token, trash.getPath())) {
        print("About to delete folder: " + fld.getPath() + "<br/>");
        OKMFolder.getInstance().purge(token, fld.getPath());
    }

    for (Document doc : OKMDocument.getInstance().getChildren(token, trash.getPath()))
        print("About to delete document: " + doc.getPath() + "<br/>");

        if (OKMDocument.getInstance().isLocked(token, doc.getPath())) {
            OKMDocument.getInstance().forceUnlock(token, doc.getPath());
        }

        OKMDocument.getInstance().purge(token, doc.getPath());
    }

    for (Mail mail : OKMMail.getInstance().getChildren(token, trash.getPath())) {
        print("About to delete mail: " + mail.getPath() + "<br/>");
        OKMMail.getInstance().purge(token, mail.getPath());
    }
}
```

Script - Put documents from folder and descendants into pending text extractio

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.openkm.api.*;
import com.openkm.bean.*;
import com.openkm.dao.*;

Logger log = LoggerFactory.getLogger("com.openkm.scripting");
int MAX_DEPTH = Integer.MAX_VALUE;

void nodeTask(String uuid, int depth) throws Exception {
    String sql = "UPDATE OKM_NODE_DOCUMENT SET NDC_TEXT_EXTRACTED='F' WHERE NBS_UUID=" + uuid;
    LegacyDAO.executeSQL(sql);

    for (Folder fld : OKMFolder.getInstance().getChildren(null, uuid)) {
        log.info("Folder: {}", fld.getPath());

        if (depth < MAX_DEPTH) {
            nodeTask(fld.getUuid(), depth + 1);
        }
    }
}

log.info("***** Process BEGIN *****");
String fldPath = "/okm:root/KnowledgeBase";
String fldUuid = OKMRepository.getInstance().getNodeUuid(null, fldPath);
nodeTask(fldUuid, 0);
log.info("***** Process END *****");
```

Script - Recursive character renaming

Renames a character in all folders or documents from some initial path.

```
// Recursive renaming & to -
import com.openkm.api.*;
import com.openkm.bean.*;

void findNodePathHelper(String path) {
    for (Folder fld : OKMFolder.getInstance().getChilds(null,path)) {
        // print (fld.getPath().substring(fld.getPath().lastIndexOf("/")).replaceAll("
        if (fld.getPath().contains("&")) {
            String newName = fld.getPath().substring(fld.getPath().lastIndexOf("/")).
            OKMFolder.getInstance().rename(null, fld.getPath(), newName);
        }

        for (Document doc : OKMDocument.getInstance().getChilds(null, fld.getPath()))
            // print (doc.getPath().substring(doc.getPath().lastIndexOf("/")).replaceA
            if (doc.getPath().contains("&")) {
                String newName = doc.getPath().substring(doc.getPath().lastIndexOf("/
                OKMDocument.getInstance().rename(null, doc.getPath(), newName);
            }
        }

        findNodePathHelper(fld.getPath());
    }
}

findNodePathHelper("/okm:root");
```


Script - Recursive repository traversal

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.openkm.api.*;
import com.openkm.bean.*;

Logger log = LoggerFactory.getLogger("com.openkm.scripting");
int MAX_DEPTH = Integer.MAX_VALUE;

void nodeTask(String path, int depth) throws Exception {
    for (Document doc : OKMDocument.getInstance().getChildren(null, path)) {
        log.info("Document: {}", doc.getPath());
    }

    for (Folder fld : OKMFolder.getInstance().getChildren(null, path)) {
        log.info("Folder: {}", fld.getPath());

        if (depth < MAX_DEPTH) {
            nodeTask(fld.getPath(), depth + 1);
        }
    }
}

log.info("***** Process BEGIN *****");
nodeTask("/okm:root", 0);
log.info("***** Process END *****");
```

Script - Recursive change security

```

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.openkm.api.*;
import com.openkm.bean.*;
import java.util.*;

Logger log = LoggerFactory.getLogger("com.openkm.scripting");
int MAX_DEPTH = Integer.MAX_VALUE;
Map grantUsers = new HashMap();
Map grantRoles = new HashMap();
Map revokeUsers = new HashMap();
Map revokeRoles = new HashMap();

void nodeTask(String uuid, int depth) throws Exception {
    setSecurity(uuid);

    for (Document doc : OKMDocument.getInstance().getChildren(null, uuid)) {
        setSecurity(doc.getUuid());
        log.info("Document: {}", doc.getPath());
    }

    for (Mail mail : OKMMail.getInstance().getChildren(null, uuid)) {
        setSecurity(mail.getUuid());
        log.info("Mail: {}", mail.getPath());
    }

    for (Folder fld : OKMFolder.getInstance().getChildren(null, uuid)) {

        if (depth < MAX_DEPTH) {
            nodeTask(fld.getUuid(), depth + 1);
        }
    }

    for (Record rec : OKMRecord.getInstance().getChildren(null, uuid)) {
        log.info("Record: {}", rec.getPath());

        if (depth < MAX_DEPTH) {
            nodeTask(rec.getUuid(), depth + 1);
        }
    }
}

void setSecurity(uuid) {
    OKMAuth.getInstance().changeSecurity(null, uuid, new HashMap(), revokeUsers, new HashMap(), grantUsers);
    OKMAuth.getInstance().changeSecurity(null, uuid, grantUsers, new HashMap(), revokeUsers, revokeRoles);
}

log.info("***** Process BEGIN *****");
String parentUuid = "234bb7b9-ecfe-4962-9245-5fd5d2bce3e3"; // Choose your folder or record

// Loading users and roles to be removed
for (String user : OKMAuth.getInstance().getUsers(null)) {
    revokeUsers.put(user, Permission.ALL);
}
for (String role : OKMAuth.getInstance().getRoles(null)) {
    revokeRoles.put(role, Permission.ALL);
}

// Loading users and roles to be added ( allowed permissions are ALL = READ + WRITE + EXECUTE )
grantRoles.put("ROLE_USER", Permission.READ);

```

```
grantUsers.put("okmAdmin", Permission.ALL);  
nodeTask(parentUuid, 0);  
log.info("***** Process END *****");
```

Script - Security change at low level

This is an administrator script that should be used only when to propagate security changes under a folder which have a significant number of nodes is needed. These type of changes made from UI can take a lot of time to be propagated across the repository and this script does it much faster.



Our recommendation is to use it when changes affects more than **100.000 nodes** otherwise we recommend do it from UI.



The security change starts with some folder UUID node and propagates across all children. The logic is to remove actual grants for each node and replace by newer based only on roles. Note that you can build other type of logic. Take the example as a basis.

- The variable **UUID** is the folder Unique Identifier.
- The variable **MAX_DEPTH** sets how much deep should advance the script in subfolders children (note than 0 indicates the UUID folder).
- The variables **roleName** and **roleGrant** arrays are the pairs of role name and roles grants to be added.
- The variable **LOG_FILE_NAME** is the name of the log file saved under \$TOMCAT_HOME folder.

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.io.Writer;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.MessageFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.sql.*;

import com.openkm.bean.Permission;
import com.openkm.core.Config;

// PARAMETERS
String LOG_FILE_NAME = "Security";
int MAX_DEPTH = Integer.MAX_VALUE;
String folder_UUID = "8ab2d963-0ced-4673-9991-54dd3d8b36b5";
String[] roleName = {"ROLE_X", "ROLE_Y"};
int[] roleGrant = {Permission.READ, Permission.READ | Permission.WRITE};

public class FileLogger {
    public static String DATE_FORMAT = "yyyy-MM-dd HH:mm:ss,SSS";
    public static String LEVEL_INFO = "INFO ";
    public static String LEVEL_WARN = "WARN ";
    public static String LEVEL_ERROR = "ERROR";

    Writer cLogger = null;

    public static void info(String baseName, String message) throws IOException {
        logWrite(baseName, LEVEL_INFO, message);
    }
}
```

```

    }

    public static void logWrite(String baseName, String level, String message) throws
        Writer sLogger = new FileWriter(getLogFile(baseName), true);
        sLogger.write(getLogEntry(level, message));
        sLogger.flush();
        sLogger.close();
    }

    public static String getLogFile(String baseName) {
        String fileDate = new SimpleDateFormat("yyyyMMdd").format(new Date());
        return Config.HOME_DIR + File.separator + baseName + "_" + fileDate + ".log";
    }

    public static void info(String message) throws IOException {
        cLogger.write(getLogEntry(LEVEL_INFO, message));
        cLogger.flush();
    }

    public static void info(String baseName, String message, String param1, String pa
        logWrite(baseName, LEVEL_INFO, message, param1, param2);
    }

    public static void warn(String baseName, String message, String param1, String pa
        logWrite(baseName, LEVEL_WARN, message, param1, param2);
    }

    public static void error(String baseName, String message, String param1, String p
        logWrite(baseName, LEVEL_ERROR, message, param1, param2);
    }

    public static void logWrite(String baseName, String level, String message, String
        Writer sLogger = new FileWriter(getLogFile(baseName), true);
        sLogger.write(getLogEntry(level, message, param1, param2));
        sLogger.flush();
        sLogger.close();
    }

    public static String getLogEntry(String level, String message) {
        StringBuilder sb = new StringBuilder();
        sb.append(new SimpleDateFormat(DATE_FORMAT).format(new Date()));
        sb.append(" ");
        sb.append(level);
        sb.append(" ");
        sb.append(message);
        sb.append("\n");
        return sb.toString();
    }

    public static String getLogEntry(String level, String message, String param1, Str
        Object[] params = {param1, param2};
        StringBuilder sb = new StringBuilder();
        sb.append(new SimpleDateFormat(DATE_FORMAT).format(new Date()));
        sb.append(" ");
        sb.append(level);
        sb.append(" ");
        sb.append(MessageFormat.format(message, params));
        sb.append("\n");
        return sb.toString();
    }
}

public void updateSecurity(Connection con, String UUID, int level) throws SQLExceptio
    FileLogger.info(LOG_FILE_NAME, "updateSecurity level: '{0}' with UUID: '{1}'",
    PreparedStatement ps = null;
    ResultSet rs = null;

```

```

    if (level == 0) {
        setParentNodeSecurity(con, UUID);
        removeAllChildrenNodeSecurity(con, UUID);
        setAllChildrenNodeSecurity(con, UUID);
    }

    if (level < MAX_DEPTH) {
        String sqlfindFolderChildren = "SELECT NBS_UUID FROM OKM_NODE_BASE WHERE NBS_

        try {
            ps = con.prepareStatement(sqlfindFolderChildren);
            ps.setString(1, UUID);
            rs = ps.executeQuery();

            while (rs.next()) {
                String childUUID = rs.getString("NBS_UUID");
                removeAllChildrenNodeSecurity(con, childUUID);
                setAllChildrenNodeSecurity(con, childUUID);
                updateSecurity(con, childUUID, level+1);
            }
        } catch (SQLException e) {
            FileLogger.error(LOG_FILE_NAME, "updateSecurity with UUID:''{0}'' con err
        } finally {
            ps.close();
            rs.close();
        }
    }
}

public void removeAllChildrenNodeSecurity(Connection con, String UUID) throws SQLExce
    PreparedStatement ps = null;
    String deleteChildrenNodeRoles = "DELETE FROM OKM_NODE_ROLE_PERMISSION where NRP_
    String deleteChildrenNodeUsers = "DELETE FROM OKM_NODE_USER_PERMISSION where NUP_

    try {
        ps = con.prepareStatement(deleteChildrenNodeRoles);
        ps.setString(1, UUID);
        ps.executeUpdate();
        ps = con.prepareStatement(deleteChildrenNodeUsers);
        ps.setString(1, UUID);
        ps.executeUpdate();
        con.commit();
    } catch (SQLException e) {
        FileLogger.error(LOG_FILE_NAME, "removeAllChildrenNodeSecurity with UUID:''{0
    } finally {
        ps.close();
    }
}

public void setAllChildrenNodeSecurity(Connection con, String UUID) throws SQLExcepti
    PreparedStatement ps = null;
    ResultSet rs = null;
    String sqlfindAllChildren = "SELECT NBS_UUID FROM OKM_NODE_BASE WHERE NBS_PARENT=
    String insertRoleSecurityChildren = "INSERT INTO OKM_NODE_ROLE_PERMISSION (NRP_NO

    try {
        ps = con.prepareStatement(sqlfindAllChildren);
        ps.setString(1, UUID);
        rs = ps.executeQuery();
        ps = con.prepareStatement(insertRoleSecurityChildren);

        while (rs.next()) {
            String childUUID = rs.getString("NBS_UUID");

            for (int i=0; i<roleName.length; i++) {
                ps.setString(1, childUUID);
                ps.setInt(2, roleGrant[i]);
            }
        }
    }
}

```

```

        ps.setString(3, roleName[i]);
        ps.addBatch();
    }
}

ps.executeBatch();
con.commit();
} catch (SQLException e) {
    FileLogger.error(LOG_FILE_NAME, "removeAllChildrenNodeSecurity with UUID:''{0}''");
} finally {
    ps.close();
    rs.close();
}
}

public void setParentNodeSecurity(Connection con, String UUID) throws SQLException, IOException {
    PreparedStatement ps = null;
    String deleteChildrenNodeRoles = "DELETE FROM OKM_NODE_ROLE_PERMISSION where NRP_NO = ?";
    String deleteChildrenNodeUsers = "DELETE FROM OKM_NODE_USER_PERMISSION where NUP_NO = ?";
    String insertRoleSecurityChildren = "INSERT INTO OKM_NODE_ROLE_PERMISSION (NRP_NO, NRP_ROLE, NRP_USER) VALUES (?, ?, ?)";

    try {
        ps = con.prepareStatement(deleteChildrenNodeRoles);
        ps.setString(1, UUID);
        ps.executeUpdate();
        ps = con.prepareStatement(deleteChildrenNodeUsers);
        ps.setString(1, UUID);
        ps.executeUpdate();
        ps = con.prepareStatement(insertRoleSecurityChildren);

        for (int i=0; i<roleName.length; i++) {
            ps.setString(1, UUID);
            ps.setInt(2, roleGrant[i]);
            ps.setString(3, roleName[i]);
            ps.addBatch();
        }

        ps.executeBatch();
        con.commit();
    } catch (SQLException e) {
        FileLogger.error(LOG_FILE_NAME, "setParentNodeSecurity with UUID:''{0}'' con");
    } finally {
        ps.close();
    }
}

FileLogger.info(LOG_FILE_NAME, "**** Security change started ****");
Class.forName("com.mysql.jdbc.Driver");
Connection con = null;
con = DriverManager.getConnection("jdbc:mysql://localhost:3306/okmdb","openkm", "****");
con.setAutoCommit(false);
updateSecurity(con, folder_UUID, 0);
con.close();
FileLogger.info(LOG_FILE_NAME, "**** Security change ended ****");

```



After the script execution we must reindex Lucene indexes because the security changes has been done at the database level and it is necessary for maintaining integrity with search engine.

Images

Initial security status

The screenshot displays the OpenKM 6.3 CE web interface. The top menu bar includes File, Edit, Tools, Bookmarks, Templates, and Help. The left sidebar shows a 'Taxonomy' tree with the path: okm:root > Pro_v12 > ContractPack_Pro_v11. The main pane shows the contents of this folder, including Documents, img, Organizer_Pro, Documentation.pdf, LICENSE.rtf, and OpenContractPack_Pro.htm. The bottom pane shows the 'Security' tab with a table of permissions for ROLE_X and ROLE_Y.

Role	Read	Write	Delete	Security	User	Read
ROLE_X	✓	✗	✗	✗	Administrator	✓
ROLE_Y	✓	✓	✗	✗	user x	✓
					user y	✓

The status bar at the bottom indicates 'Connected as Administrator' and shows a file size of 80.5 MB.

UUID of the folder

Path : /okm:root/Pro_v12/ContractPack_Pro_v11/

	Name	Size	Update date	Author
<input type="checkbox"/>	Documents		14-03-2013 10:17:31	Admini
<input type="checkbox"/>	img		14-03-2013 10:18:51	Admini
<input type="checkbox"/>	Organizer_Pro		14-03-2013 10:18:11	Admini
<input type="checkbox"/>	Documentation.pdf	248.4 KB	14-03-2013 10:18:11	Admini
<input type="checkbox"/>	LICENSE.rtf	37.8 KB	14-03-2013 10:17:31	Admini
<input type="checkbox"/>	OpenContractPack_Pro.htm	4.7 KB	14-03-2013 10:18:11	Admini

Properties | Notes | Security

UUID 86d931e1-339a-4b74-8b8d-6a01a40ca6ad

Name ContractPack_Pro_v11

Parent /okm:root/Pro_v12

Created 14-03-2013 10:17:36 by Administrator

Subscribed No

Folders 3

Documents 3

Mails 0

Keywords

URL

WebDAV

Subscribed users

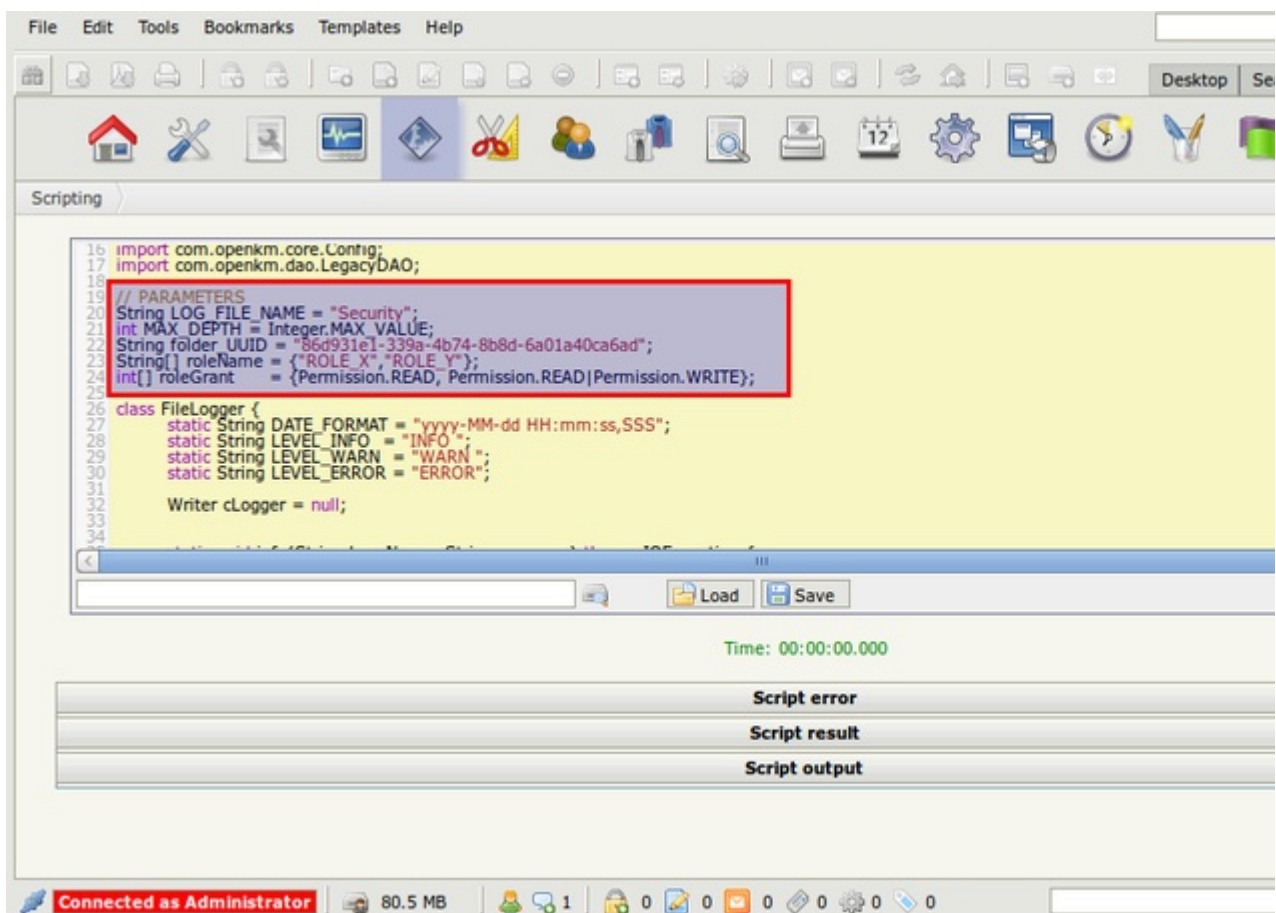
Keywords cloud

Categories

Connected as Administrator 80.5 MB 1 0 0 0 0 0 0 0 0

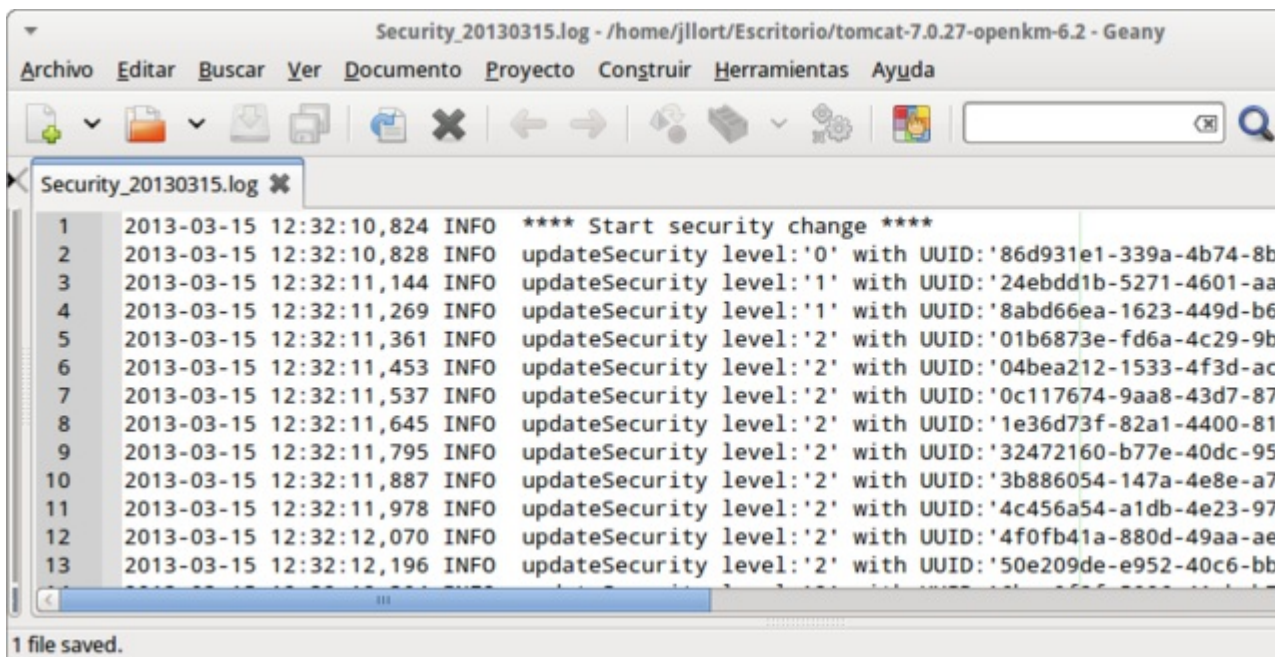
Execution

Change script parameters by your owns.



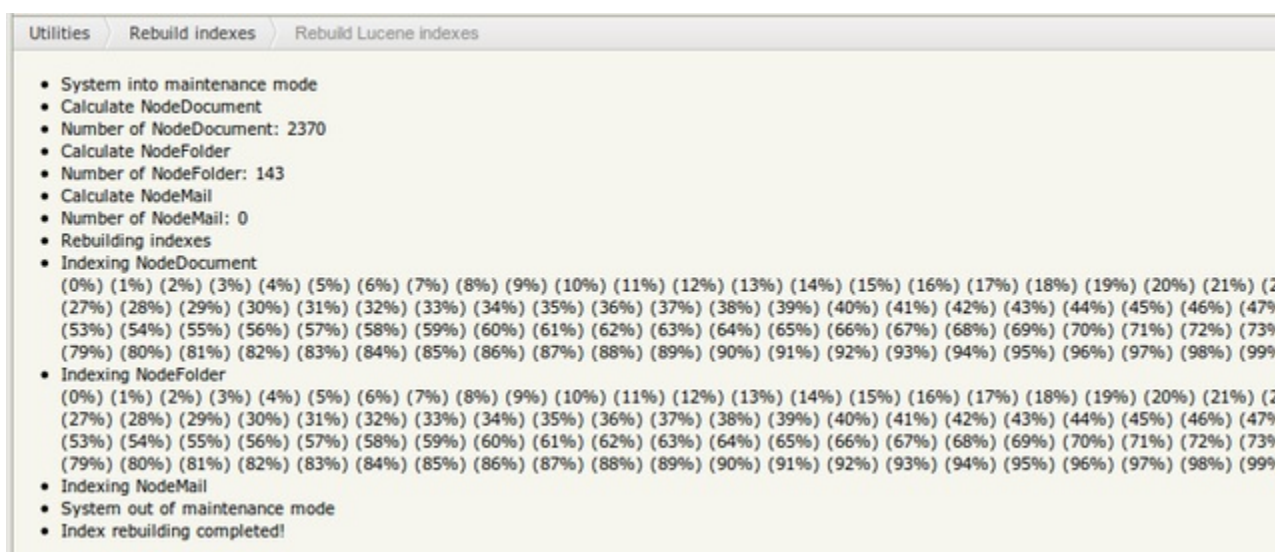
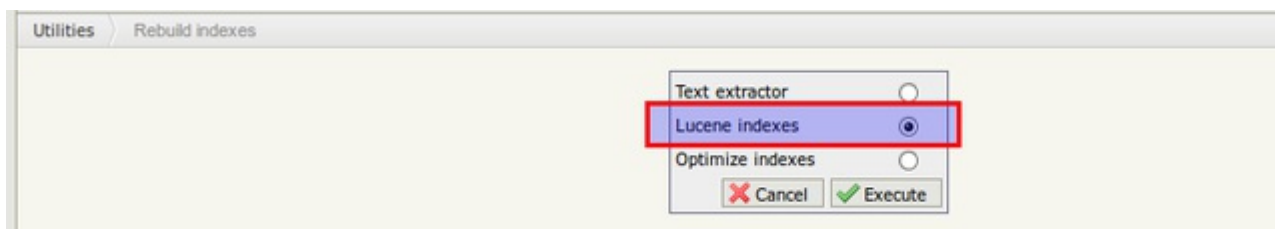
Log

Consider take a look at file log to see progress of the process and check for errors.



Rebuild Lucene indexes

Caused by changes of the security at a database level, it is necessary for maintaining integrity with search engine, reindex Lucene indexes.



Security changed applied

The screenshot displays the OpenKM 6.3 CE web interface. The top menu bar includes File, Edit, Tools, Bookmarks, Templates, and Help. The left sidebar shows a 'Taxonomy' tree with the path: okm:root > Pro_v12 > ContractPack_Pro_v11. The main panel shows the file list for the path: /okm:root/Pro_v12/ContractPack_Pro_v11/. The file list table is as follows:

	Name	Size	Update date	Author
<input type="checkbox"/>	Documents		14-03-2013 10:17:31	Admini
<input type="checkbox"/>	img		14-03-2013 10:18:51	Admini
<input type="checkbox"/>	Organizer_Pro		14-03-2013 10:18:11	Admini
<input type="checkbox"/>	Documentation.pdf	248.4 KB	14-03-2013 10:18:11	Admini
<input type="checkbox"/>	LICENSE.rtf	37.8 KB	14-03-2013 10:17:31	Admini
<input type="checkbox"/>	OpenContractPack_Pro.htm	4.7 KB	14-03-2013 10:18:11	Admini

Below the file list, the 'Security' tab is active, showing a table of permissions for two roles:

Role	Read	Write	Delete	Security	Update	User	Read
ROLE_X	✓	✗	✗	✗			
ROLE_Y	✓	✓	✗	✗			

The bottom status bar indicates 'Connected as Administrator', '80.5 MB' of storage, and various system icons.

Script samples for automation

Script	Description
<i>Script for automation - Autonumeric</i>	<i>Sets an unique number for each uploaded document.</i>
<i>Script for automation - Execute OMR</i>	<i>Executes the OMR engine for each new uploaded image.</i>
<i>Script for automation - Set unique document name based on metadata</i>	<i>Each time a document is uploaded the name of the document is changed based on a metadata group values.</i>
<i>Script for automation - Simple auto tagging</i>	<i>Auto tagging a new document based on a restricted list is defined on database metadata tables.</i>

Script for automation - Autonumeric

Creates an unique number for each new document.



- There's a metadata with tree fields which stores data.
- Each time a new document is uploaded, automation is executed. Automation script creates a new unique number identifier using metadata database sequence feature.
- The **document id** is a 6 digits incremental numeric
- The **revision** number starts with 1
- The **code** is a unique id + "-" + revision



To get it running it's needed to register a metadata group and then create an automation task based on scripting.

Metadata group definition:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.1//EN"
"http://www.openkm.com/dtd/property-groups-2.1.dtd">
<property-groups>
  <property-group label="Autonumeric" name="okg:autonumber">
    <input label="Document ID" name="okp:autonumber.id" type="text" readonly="true"/>
    <input label="Revision" name="okp:autonumber.revision" type="text" readonly="true"/>
    <input label="Code" name="okp:autonumber.code" type="text" readonly="true"/>
  </property-group>
</property-groups>
```

The automation script:

```
import com.openkm.api.OKMPropertyGroup;
import com.openkm.api.OKMRepository;
import java.util.*;
import com.openkm.dao.DatabaseMetadataDAO;

String grpName = "okg:autonumber";
String table = "autonumber";
String sequenceName = "doc_id";
String path = OKMRepository.getInstance().getNodePath(null, uuid);

// Add Group
OKMPropertyGroup.getInstance().addGroup(null, path, grpName);

// Setting properties
String id = String.valueOf(DatabaseMetadataDAO.getNextSequenceValue(table, sequenceName));

switch (id.length()) {
  case 1:
    id = "00000" + id;
    break;
  case 2:
    id = "0000" + id;
    break;
  case 3:
    id = "000" + id;
    break;
  case 4:
    id = "00" + id;
    break;
  case 5:
    id = "0" + id;
    break;
}
```

```

        id = "0000" + id;
        break;
    case 3:
        id = "000" + id;
        break;
    case 4:
        id = "00" + id;
        break;
    case 5:
        id = "0" + id;
        break;
    }

    String revision = "1";
    String code = id + "-" + revision;

    Map map = new HashMap();
    map.put("okp:autonumber.id", id);
    map.put("okp:autonumber.revision", revision);
    map.put("okp:autonumber.code", code);
    OKMPropertyGroup.getInstance().setPropertiesSimple(null, path, grpName, map);

```

Images

Register metadata group

Property groups						
Group label		Group name				
Autonumeric		okp:autonumber				
Label	Name	Width	Height	Field		
Document ID	okp:autonumber.id	100px	25px	Input	Readonly: true Data: Type: text	
Revision	okp:autonumber.revision	100px	25px	Input	Readonly: true Data: Type: text	
Code	okp:autonumber.code	100px	25px	Input	Readonly: true Data: Type: text	

Register automation rule

Automation rules							
#	Order	Name	Event	At	Validations	Actions	Exclusive
1	1	Autonumeric	doc_create	post	1	1	✗

Automation rules > Automation definition

Add validation: - +

VALIDATIONS				
Order	Type	active	Param0	Param1
1	PathContains	<input checked="" type="checkbox"/>	String:/okm:root	

Add action: ExecuteScripting +

Create action

Name ExecuteScripting

Group action

ClassName com.openkm.automation.action.ExecuteScripting

Active ☒

Parameters:

Order 1

Script


```


1 import com.openkm.api.OKMPropertyGroup;
2 import com.openkm.api.OKMRepository;
3 import java.util.*;
4 import com.openkm.dao.DatabaseMetadataDAO;
5
6 String grpName = "okg:autonumber";
7 String table = "autonumber";
8 String sequenceName = "doc_id";
9 String path = OKMRepository.getInstance().getNodePath(null,uuid);
10 // Add Group
11 OKMPropertyGroup.getInstance().addGroup(null, path, grpName);
12 // Setting properties
13 String id = String.valueOf(DatabaseMetadataDAO.getNextSequenceValue(table, sequenceName));
14 switch (id.length()) {
15     case 1:
16         id = "00000" + id;
17         break;
18     case 2:
19         id = "0000" + id;
20         break;
21     case 3:
22         id = "000" + id;
23         break;
24     case 4:
25         id = "00" + id;
26         break;
27     case 5:
28         id = "0" + id;
29         break;
30 }


```


ACTIONS				
Order	Type	active	Param0	Param1

Automation rules > Automation definition





Add validation: 

VALIDATIONS				
Order	Type	active	Param0	Pa
1	PathContains		String:/okm:root	

Add action: 

ACTIONS				
Order	Type	active	Param0	Pa
1	ExecuteScripting			

Result after uploading a new document

		Name	Size	Update date	Version
<input type="checkbox"/>		2012		09-11-2012 09:22:15	
<input type="checkbox"/>		access.sh	5.0 KB	16-11-2012 11:34:45	1.0
<input type="checkbox"/>		logo_mobile.gif	3.4 KB	16-11-2012 12:27:10	1.0
<input type="checkbox"/>		mail.txt	2.0 KB	05-01-2013 20:32:03	1.0

Properties	Notes	Security	History	Preview	Autonumeric
					<input type="button" value="Change"/> <input type="button" value="Delete"/>
Document ID 000004 Revision 1 Code 000004-1					

Script for automation - Execute OMR

The script process an image with OMR engine from Automation each time new document is uploaded. With minimal changes can be used in conjunction with crontab.



The value of the variable **omId** is the **template id**.

```
import com.openkm.bean.Document;
import com.openkm.module.db.stuff.DbSessionManager;
import com.openkm.api.OKMDocument;
import com.openkm.api.OKMRepository;
import com.openkm.omr.OMRHelper;

long omId= 4;
String systemToken = DbSessionManager.getInstance().getSystemToken();
String docPath = OKMRepository.getInstance().getNodePath(systemToken, uuid);
Document doc = OKMDocument.getInstance().getProperties(systemToken, docPath);

if (OMRHelper.isValid(doc)) {
    OMRHelper.processAndStoreMetadata(omId, uuid);
}
```

Images

Register automation rule

#	Order	Name	Event	At	Validations	Actions	Ex
1	0	omr test	doc_create	post	1	1	

With PathContains and ExecuteScripting action.

Add validation: <input type="text" value="-"/>			
VALIDATIONS			
Order	Type	active	Param0
1	PathContains		String:/okm:root/test2
Add action: <input type="text" value="-"/>			
ACTIONS			
Order	Type	active	Param0
1	ExecuteScripting		

The script code:

Edit action	
Name	ExecuteScripting
Group	action
ClassName	com.openkm.automation.action.ExecuteScripting
Active	<input checked="" type="checkbox"/>
Parameters:	
Order	1
Script	<pre> 1 import com.openkm.bean.Document; 2 import com.openkm.module.db.stuff.DbSessionManager; 3 import com.openkm.api.OKMDocument; 4 import com.openkm.api.OKMRepository; 5 import com.openkm.omr.OMRHelper; 6 7 long omId= 4; 8 String systemToken = DbSessionManager.getInstance().getSystemToken(); 9 String docPath = OKMRepository.getInstance().getNodePath(systemToken, uuid); 10 Document doc = OKMDocument.getInstance().getProperties(systemToken, docPath); 11 if (OMRHelper.isValid(doc)) { 12 OMRHelper.processAndStoreMetadata(omId, uuid); 13 } </pre>
Cancel	

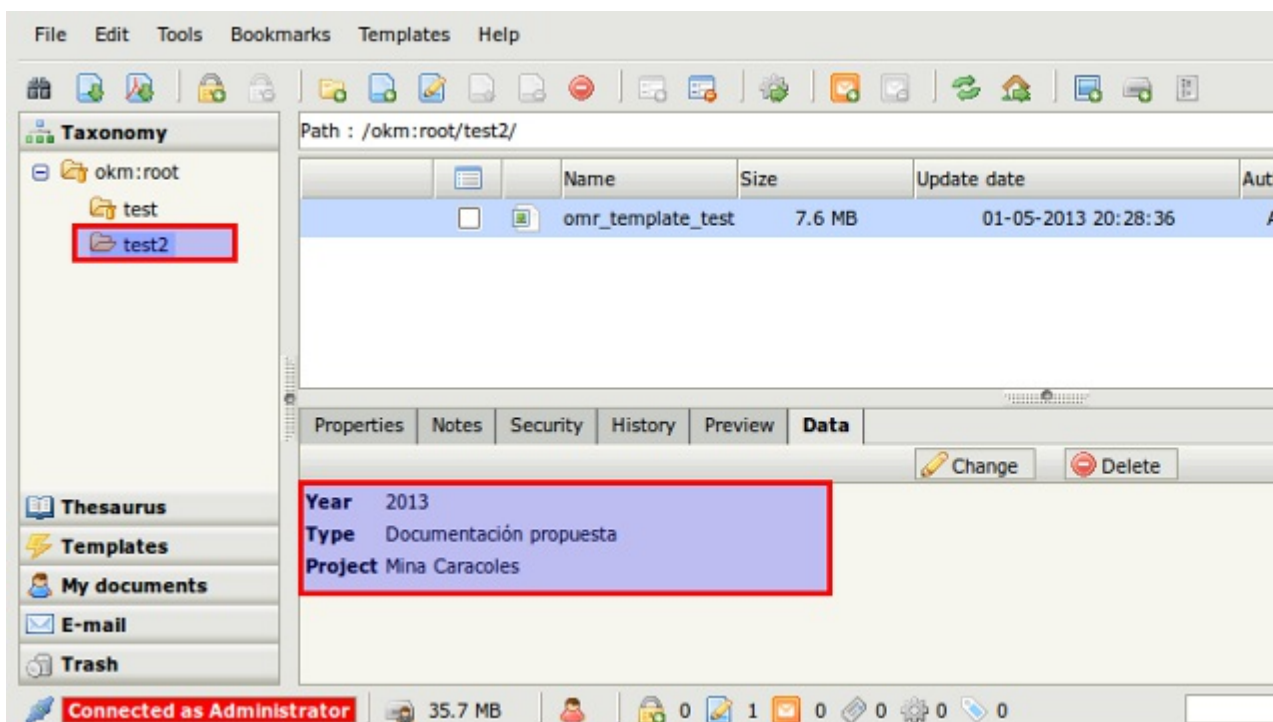
Templates definition

Get the **omId** value of the **template**.

Omr Template					
Id	Name	Template	Asc	Config	
4	Template test	template.png	template.png.asc	template.png.config	templat
6	Template test2	2circle-org-colored-whole.tif	2circle-org-colored-whole.tif.asc	2circle-org-colored-whole.tif.config	2circle-

Result

Each document uploaded into folder path (or subfolders), when is valid image file, is processed by OMR engine.



The screenshot displays the OpenKM 6.3 CE web interface. On the left, the 'Taxonomy' sidebar shows a tree structure with 'okm:root', 'test', and 'test2' (highlighted with a red box). Below the taxonomy are links for 'Thesaurus', 'Templates', 'My documents', 'E-mail', and 'Trash'. The main content area shows the path '/okm:root/test2/'. A table lists the file 'omr_template_test' with a size of 7.6 MB and an update date of 01-05-2013 20:28:36. Below the table, the 'Data' tab is active, showing the following metadata:

Year	2013
Type	Documentación propuesta
Project	Mina Caracoles

Buttons for 'Change' and 'Delete' are visible next to the metadata. The bottom status bar indicates 'Connected as Administrator' and shows a storage usage of 35.7 MB.

Script for automation - Set unique document name based on metadata

Each time a new document is uploaded, the script sets a unique document name based in metadata values.



- Property **okp:data.id** stores a unique auto incremental value.
- Property **okp:data.project.code** stores a project code value.
- Property **okp:data.customer** stores a customer code.
- Property **okp:data.description** stores a document description.
- When a new document is uploaded the user must fill all the fields except **okp:data.id** which is automatically set by OpenKM.
- When Metadata group is changed - event triggered- is executed automation code which generates **okp:data.id** and renames the document based on mask **projectCode-autonumericId-clientCode-description.documentExtension**.

Metadata group definition:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.1//EN"
"http://www.openkm.com/dtd/property-groups-2.1.dtd">
<property-groups>
  <property-group label="Datos" name="okp:data">
    <input label="Id" type="text" name="okp:data.id" width="200px" readonly="true" />
    <input label="Project code" type="text" name="okp:data.project.code" width="200px" />
    <validator type="req"/>
    <validator type="num"/>
    <validator type="maxlen" parameter="6"/>
    <validator type="minlen" parameter="6"/>
  </input>
  <select label="Customer" name="okp:data.customer" type="simple" table="customer" />
  <optionsQuery="select $cus_id, $cus_nombre from DatabaseMetadataValue where $cus_nombre like '%$%'"/>
  <validator type="req"/>
</select>
  <input label="Description" type="text" name="okp:data.description" width="200px" />
  <validator type="req"/>
  <validator type="maxlen" parameter="150"/>
</input>
</property-group>
</property-groups>
```

Database metadata definition:

```
-- metadata type
DELETE FROM OKM_DB_METADATA_TYPE WHERE DMT_TABLE='customer';
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN) VALUES ('customer', 'id', 'text', 'virtual');
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN) VALUES ('customer', 'project_code', 'text', 'virtual');
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN) VALUES ('customer', 'customer_code', 'text', 'virtual');
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN) VALUES ('customer', 'description', 'text', 'virtual');

-- values
DELETE FROM OKM_DB_METADATA_VALUE WHERE DMV_TABLE='customer';
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('customer', 'id', '1');
```

```
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('customer
```

The script:

```
import com.openkm.api.OKMPropertyGroup;
import com.openkm.api.OKMRepository;
import java.util.*;
import com.openkm.dao.DatabaseMetadataDAO;
import com.openkm.api.OKMPropertyGroup;
import com.openkm.bean.form.FormElement;
import com.openkm.bean.form.Input;
import com.openkm.bean.form.Select;
import com.openkm.bean.form.Option;
import com.openkm.util.PathUtils;
import com.openkm.util.FileUtils;
import com.openkm.api.OKMDocument;

String grpName = "okg:data";
String table = "autonumber";
String sequenceName = "doc_id";
String path = OKMRepository.getInstance().getNodePath(null,uuid);

// Evaluate if already has property group
boolean add = true;
String prjCode = "";
String clientCode = "";
String docId = "";
String desc = "";

for (FormElement formElement : OKMPropertyGroup.getInstance().getProperties(null, path)) {
    if (formElement.getName().equals("okp:data.id")) {
        docId = ((Input) formElement).getValue();
        add = docId.equals("");
    } else if (formElement.getName().equals("okp:data.project.code")) {
        prjCode = ((Input) formElement).getValue();
    } else if (formElement.getName().equals("okp:data.description")) {
        desc = ((Input) formElement).getValue();
    } else if (formElement.getName().equals("okp:data.customer")) {
        for (Option option: ((Select) formElement).getOptions()) {
            if (option.isSelected()) {
                clientCode = option.getValue();
            }
        }
    }
}

// Setting properties
if (add) {
    // add unique document id
    docId = String.valueOf(DatabaseMetadataDAO.getNextSequenceValue(table, sequenceName));

    switch (docId.length()) {
        case 1:
            docId = "0000" + docId;
            break;
        case 2:
            docId = "000" + docId;
            break;
        case 3:
            docId = "00" + docId;
            break;
        case 4:
            docId = "0" + docId;
            break;
    }
}
```

```

    }

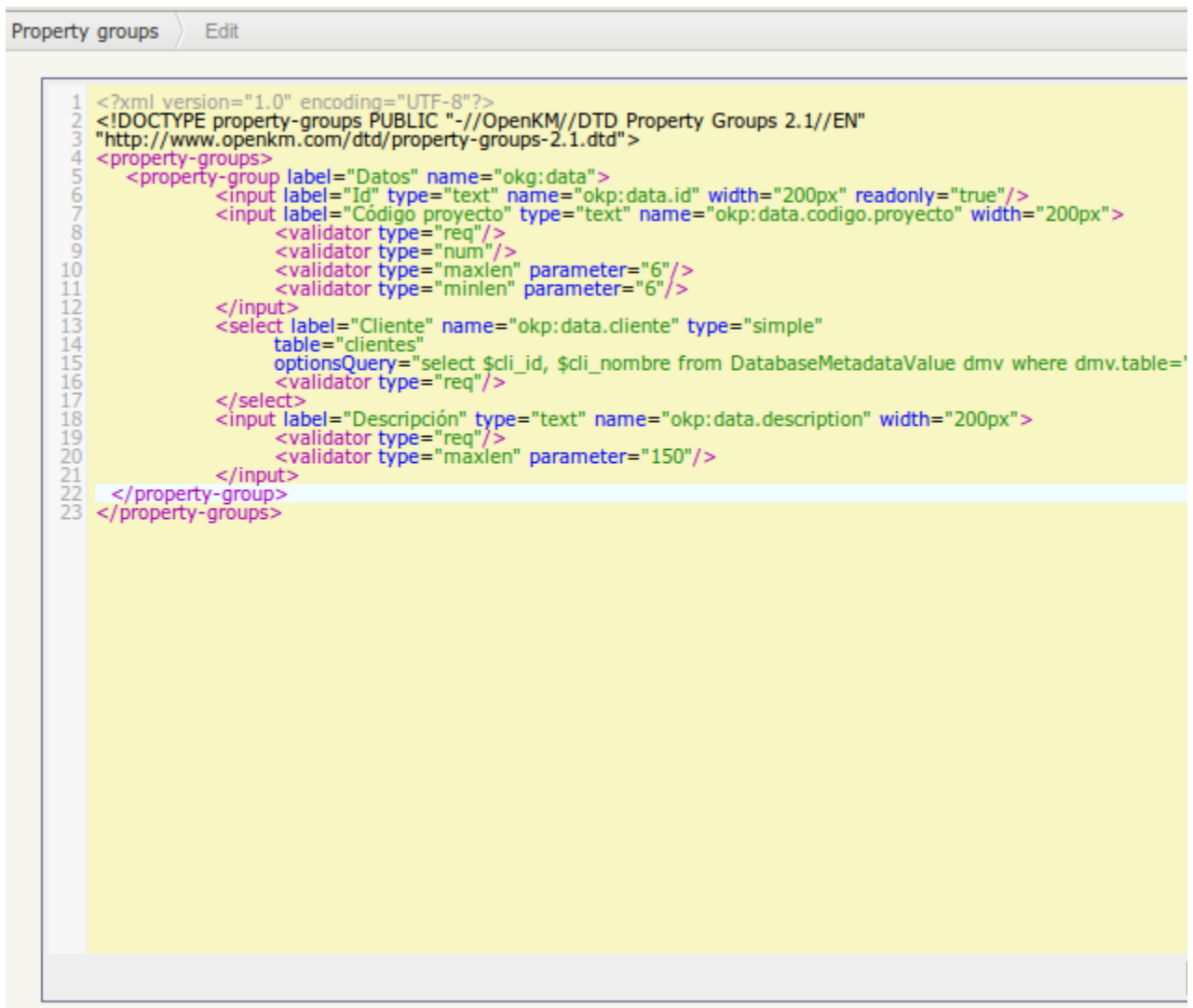
    Map map = new HashMap();
    map.put("okp:data.id", docId);
    OKMPropertyGroup.getInstance().setPropertiesSimple(null, path, grpName, map);
}

// rename document
String newName = prjCode + "-" + docId + "-" + clientCode + "-" + desc + "." + FileUt.
OKMDocument.getInstance().rename(null, path, newName);

```


Images


Register metadata group





Register automation rule

Create automation task to show property group in document wizard:

Add validation: 

VALIDATIONS			
Order	Type	active	Param0
1	PathContains		String:/okm:root

Add action: 

ACTIONS			
Order	Type	active	Param0
1	AddPropertyGroupToWizard		Property group:okg:data

Create automation task based on `property_group_set` event:

#	Order	Name	Event	At	Validations	Actions
1	1	Autonumeric	doc_create	post	1	1
2	2	Autonumeric	prop_group_set	post	1	1

Add validation: - +

VALIDATIONS			
Order	Type	active	Param0
1	PathContains		String:/okm:root

Add action: - +

Edit action

Name ExecuteScripting
Group action
ClassName com.openkm.automation.action.ExecuteScripting
Active ☒
Parameters:
Order 1

Script

```

1 import com.openkm.api.OKMPropertyGroup;
2 import com.openkm.api.OKMRepository;
3 import java.util.*;
4 import com.openkm.dao.DatabaseMetadataDAO;
5 import com.openkm.api.OKMPropertyGroup;
6 import com.openkm.bean.form.FormElement;
7 import com.openkm.bean.form.Input;
8 import com.openkm.bean.form.Select;
9 import com.openkm.bean.form.Option;
10 import com.openkm.util.PathUtils;
11 import com.openkm.util.FileUtils;
12 import com.openkm.api.OKMDocument;
13
14 String grpName = "okg:data";
15 String table = "autonumber";
16 String sequenceName = "doc_id";
17 String path = OKMRepository.getInstance().getNodePath(null,uuid);
18
19 // Evaluate if already has property group
20 boolean add = true;
21 String prjCode = "";
22 String clientCode = "";
23 String docId = "";
24 String desc = "";
25 for (FormElement formElement : OKMPropertyGroup.getInstance().getProperties(nul
26     if (formElement.getName().equals("okp:data.id")) {
27         docId = ((Input) formElement).getValue();
28         add = docId.equals("");


```

Results


Upload a new file:


Document wizard: scaled.png

Id

Project code 


Customer

Description 



 **Accept**

Automatically the document is created based on metadata fields values:

Path : /okm:root/test/

	Name	Size	Update date
	321456-00011-UGT-Test.png	160.0 KB	02-05-2013 13:5

Properties Notes Security History Preview Wiki **Datos**

 **Change**  **Delete**

Id 00011
Project code 321456
Customer UGT
Description Test

Script for automation - Simple auto tagging

The script sets documents keywords. The keywords in the database metadata are used to tag the document, each keyword present in a document content is considered a candidate.



- There's a database metadata table named **doc_type** with allowed keywords values.
- There's a **script** that parses contents words, looking for coincidences with keywords stored into database metadata.
- There's an **automation task** - based on scripting - executed after each uploaded document that process each document.

Database metadata definition:

```
-- DOCS TYPE
DELETE FROM OKM_DB_METADATA_TYPE WHERE DMT_TABLE='doc_type';
INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN)
VALUES ('doc_type', 'doc_type', 'doc_type', 'doc_type');

-- VALUES
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COLUMN, DMV_COLUMN) VALUES ('doc_type', 'doc_type', 'doc_type');
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COLUMN, DMV_COLUMN) VALUES ('doc_type', 'doc_type', 'doc_type');
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COLUMN, DMV_COLUMN) VALUES ('doc_type', 'doc_type', 'doc_type');
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COLUMN, DMV_COLUMN) VALUES ('doc_type', 'doc_type', 'doc_type');
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COLUMN, DMV_COLUMN) VALUES ('doc_type', 'doc_type', 'doc_type');
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COLUMN, DMV_COLUMN) VALUES ('doc_type', 'doc_type', 'doc_type');
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COLUMN, DMV_COLUMN) VALUES ('doc_type', 'doc_type', 'doc_type');
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COLUMN, DMV_COLUMN) VALUES ('doc_type', 'doc_type', 'doc_type');
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COLUMN, DMV_COLUMN) VALUES ('doc_type', 'doc_type', 'doc_type');
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COLUMN, DMV_COLUMN) VALUES ('doc_type', 'doc_type', 'doc_type');
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COLUMN, DMV_COLUMN) VALUES ('doc_type', 'doc_type', 'doc_type');
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COLUMN, DMV_COLUMN) VALUES ('doc_type', 'doc_type', 'doc_type');
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COLUMN, DMV_COLUMN) VALUES ('doc_type', 'doc_type', 'doc_type');
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COLUMN, DMV_COLUMN) VALUES ('doc_type', 'doc_type', 'doc_type');
INSERT INTO OKM_DB_METADATA_VALUE (DMV_TABLE, DMV_COLUMN, DMV_COLUMN) VALUES ('doc_type', 'doc_type', 'doc_type');
```

The script:

```
import com.openkm.dao.bean.NodeDocumentVersion;
import com.openkm.dao.bean.NodeDocument;
import com.openkm.extractor.TextExtractorWork;
import com.openkm.api.OKMRepository;
import com.openkm.dao.NodeDocumentVersionDAO;
import com.openkm.dao.NodeDocumentDAO;
import org.hibernate.Session;
import com.openkm.dao.HibernateUtil;
import org.hibernate.Query;
import com.openkm.dao.bean.DatabaseMetadataValue;
import com.openkm.module.db.stuff.DbSessionManager;
import com.openkm.api.OKMProperty;
import com.openkm.dao.HibernateUtil;
import com.openkm.core.DatabaseException;
```

```

import org.hibernate.HibernateException;

String systemToken = DbSessionManager.getInstance().getSystemToken();

// Get path
String docPath = OKMRepository.getInstance().getNodePath(null, uuid);

// Get doc version uuid
NodeDocumentVersion currentVersion = NodeDocumentVersionDAO.getInstance().findCurrent
String docVerUuid = currentVersion.getUuid();

// Document extractor
TextExtractorWork tew = new TextExtractorWork();
tew.setDocUuid(uuid);
tew.setDocPath(docPath);
tew.setDocVerUuid(docVerUuid);

// Execute extractor
NodeDocumentDAO.getInstance().textExtractorHelper(tew);

// Get extracted text
NodeDocument docNode = NodeDocumentDAO.getInstance().findByPk(uuid);
String text = docNode.getText().toLowerCase();

// Looking for metadata description values
String qs = "from DatabaseMetadataValue";
Session session = HibernateUtil.getSessionFactory().openSession();

try {
    Query q = session.createQuery(qs);
    List ret = q.list();

    for (DatabaseMetadataValue dmv : ret) {
        if (text.contains(dmv.getCol01().toLowerCase())) {
            OKMProperty.getInstance().addKeyword(systemToken, docPath, dmv.getCol01().toLow
        }
    }
} catch (HibernateException e) {
    throw new DatabaseException(e.getMessage(), e);
} finally {
    HibernateUtil.close(session);
}

```

Images

Register metadata database definition

Database query

```

1 -- DOCS TYPE
2 DELETE FROM OKM_DB_METADATA_TYPE WHERE DMT_TABLE='doc_type';
3 INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN) V
4 INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN) V
5
6 -- VALUES
7 INSERT INTO OKM_DB_METADATA_TYPE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('doc_type', '1', 'Article
8 INSERT INTO OKM_DB_METADATA_TYPE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('doc_type', '25', 'Audio
9 INSERT INTO OKM_DB_METADATA_TYPE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('doc_type', '6', 'Broke
10 INSERT INTO OKM_DB_METADATA_TYPE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('doc_type', '8', 'Case
11 INSERT INTO OKM_DB_METADATA_TYPE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('doc_type', '5', 'Compi
12 INSERT INTO OKM_DB_METADATA_TYPE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('doc_type', '4', 'Confe
13 INSERT INTO OKM_DB_METADATA_TYPE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('doc_type', '3', 'Cours
14 INSERT INTO OKM_DB_METADATA_TYPE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('doc_type', '7', 'Disser
15 INSERT INTO OKM_DB_METADATA_TYPE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('doc_type', '12', 'Form
16 INSERT INTO OKM_DB_METADATA_TYPE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('doc_type', '11', 'Imag
17 INSERT INTO OKM_DB_METADATA_TYPE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('doc_type', '26', 'Infog
18 INSERT INTO OKM_DB_METADATA_TYPE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('doc_type', '16', 'Inter
19 INSERT INTO OKM_DB_METADATA_TYPE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('doc_type', '17', 'Pres
20 INSERT INTO OKM_DB_METADATA_TYPE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('doc_type', '19', 'Repo
21 INSERT INTO OKM_DB_METADATA_TYPE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('doc_type', '14', 'Vide
22 INSERT INTO OKM_DB_METADATA_TYPE (DMV_TABLE, DMV_COL00, DMV_COL01) VALUES ('doc_type', '0', '_Othe

```

Tables

```

DELETE FROM OKM_DB_METADATA_TYPE WHERE DMT_TABLE='doc

Time: 00:00:00.000
Row Count: 0

INSERT INTO OKM_DB_METADATA_TYPE (DMT_TABLE, DMT_REAL_COLUMN, DMT_TYPE, DMT_VIRTUAL_COLUMN) V

Time: 00:00:00.001
Row Count: 1

```

Register automation rule

Automation rules [Create automation rule](#)

Order	<input type="text" value="1"/>
Name	<input type="text" value="Autotaging based on metadata"/>
Event	<input type="text" value="Document creation"/>
At	<input type="text" value="post"/>
Exclusive	<input type="checkbox"/>
Active	<input checked="" type="checkbox"/>

Automation rules

#	Order	Name	Event	At	Validations	Action
1	1	Autotaging based on metadata	Document creation	post	0	0

Automation rules Automation definition

Automation rule Autotaging based on metadata

Add validation: -

VALIDATIONS				
Order	Type	active	Param0	Param1
1	PathContains	<input checked="" type="checkbox"/>	String:/okm:root	

Add action: -

Edit action

Name ExecuteScripting

Group action

ClassName com.openkm.automation.action.ExecuteScripting

Active ☒

Parameters:

Order

Script

```

1 import com.openkm.dao.bean.NodeDocumentVersion;
2 import com.openkm.dao.bean.NodeDocument;
3 import com.openkm.extractor.TextExtractorWork;
4 import com.openkm.api.OKMRepository;
5 import com.openkm.dao.NodeDocumentVersionDAO;
6 import com.openkm.dao.NodeDocumentDAO;
7 import org.hibernate.Session;
8 import com.openkm.dao.HibernateUtil;
9 import org.hibernate.Query;
10 import com.openkm.dao.bean.DatabaseMetadataValue;
11 import com.openkm.module.db.stuff.DbSessionManager;
12 import com.openkm.api.OKMProperty;
13 import com.openkm.dao.HibernateUtil;
14 import com.openkm.core.DatabaseException;
15 import org.hibernate.HibernateException;
16
17 String systemToken = DbSessionManager.getInstance().getSystemToken();
18 //String uuid = "7f8b48ee-5efd-48dc-82ec-70053f8ab709";
19 // Getting path
20 String docPath = OKMRepository.getInstance().getNodePath(null, uuid);
21 // Getting doc version uuid
22 NodeDocumentVersion currentVersion = NodeDocumentVersionDAO.getInstance().findCurrentVersion(docPath);
23 String docVerUuid = currentVersion.getUuid();
24 // Document extractor
25 TextExtractorWork tew = new TextExtractorWork();
26 tew.setDocUuid(uuid);
27 tew.setDocPath(docPath);
28 tew.setDocVerUuid(docVerUuid);
29 //tew.setTenant(1);
30 // Execute extractor

```

ACTIONS				
Order	Type	active	Param0	Param1
1	ExecuteScripting	<input checked="" type="checkbox"/>		

Result

Path : /okm:root/

☒ Paginate 10 Total: 4 1 to 4 Order by Reverse

	Name	Title	Language	Size	Update c
<input type="checkbox"/>	facturas				24-01-20
<input type="checkbox"/>	gwt				17-05-20
<input type="checkbox"/>	records				08-04-20
<input type="checkbox"/>	BBC - giro canceled.txt		inglés	2.9 KB	24-05-20

Properties Notes Security History Preview Activity log Workflow

UUID f7aefe1d-3bd2-49ea-9d6d-c0e66c3dd191

Name BBC - giro canceled.txt

Title

Folder /okm:root

Size 2.9 KB

Version 1.0

Language inglés

Created 24-05-2013 13:36:12 by Administrator

Modified 24-05-2013 13:36:12 by Administrator

MIME type text/plain

Keywords

Keywords cloud article

Categories

Status Available

Subscribed No

History size 2.9 KB

URL

WebDAV

Crontab samples

<i>Script</i>	<i>Description</i>
<i>Crontab sample - Basic file system document importer</i>	<i>The script imports files from the server file system.</i>
<i>Crontab sample - Document importer with metadata values</i>	<i>Imports files with metadata values from the server file system.</i>

Crontab sample - Basic file system document importer

The script import files from file system of the server.



- Files are imported from the server file system.
- Files are periodically imported by a crontab task to a OpenKM folder (/okm:root/Scans).

```
import com.openkm.core.*;
import com.openkm.api.*;
import java.io.*;
import com.openkm.module.db.stuff.DbSessionManager;

String token = DbSessionManager.getInstance().getSystemToken();
OKMDocument document = OKMDocument.getInstance();
OKMFolder folder = OKMFolder.getInstance();

public void autoImport(String okmPath, File fldpath){
    try {
        print("Scanning " + fldpath.getName() + "<br>");
        for (File file : fldpath.listFiles()) {
            print("Importing " + file.getName() + "<br>");
            try {
                if (file.isDirectory()) {
                    try {
                        folder.createSimple(token, okmPath + file.getName());
                    } catch (ItemExistsException ie) {
                        print("folder already exists<br>");
                        // Folder already exists - just ignore exception
                    }
                    autoImport( okmPath + file.getName() + "/", file);
                } else {
                    // Check if file is still being written to
                    long length = file.length();
                    Thread.sleep(1000);
                    if (file.length() > length) continue; // Skip file this time
                    FileInputStream fis = new FileInputStream(file);
                    document.createSimple(token, okmPath + file.getName(), fis);
                    fis.close();
                }
                print("Created " + okmPath + file.getName() + "<br>");
            } catch (Exception e) {
                print ("Exception:" + e + "<br>");
                // Something bad happened to prevent import. Skip to next file.
                continue;
            }
            file.delete();
        }
    } catch (Exception e) {
        print("Exception: " + e + "<br>");
    }
}










autoImport("/okm:root/Scans/", new File("/home/fom/scanner"));
```

Images











Register a crontab task

Crontab					
Name	Expression	Mime	File Name	Mail	L
Dashboard Activity Purge	@daily	application/x-bsh	DashboardActivityPurge.bsh	noreply@openkm.com	17-ene
User Items Serialize	@hourly	application/x-bsh	UserItemsSerialize.bsh	noreply@openkm.com	24-ene
User Mail Importer	*/30 * * * *	application/x-bsh	UserMailImporter.bsh	noreply@openkm.com	24-ene
Text Extractor Worker	*/5 * * * *	application/x-bsh	TextExtractorWorker.bsh	noreply@openkm.com	25-ene
Repository Info	@daily	application/x-bsh	RepositoryInfo.bsh	noreply@openkm.com	17-ene
Session Watchdog	*/5 * * * *	application/x-bsh	SessionWatchdog.bsh	noreply@openkm.com	25-ene
Process Pending Tasks	*/5 * * * *	application/x-bsh	ProcessPendingTasks.bsh	noreply@openkm.com	25-ene
Simple importer	0 6 * * *	application/x-bsh	simple_importer.bsh		

Files in the application server

		
	CI0477978849.pdf	244,2 KB Adobe Acrobat Document 29/09/11
	CI0484641433.pdf	270,3 KB Adobe Acrobat Document 29/09/11
	CI0491320030.pdf	276,4 KB Adobe Acrobat Document 29/09/11
	CI0497971496.pdf	279,0 KB Adobe Acrobat Document 29/09/11
	CI0504591121.pdf	278,9 KB Adobe Acrobat Document 29/09/11
	CI0517932793.pdf	305,1 KB Adobe Acrobat Document 29/09/11

Result

Taxonomy		Path : /okm:root/Scans/			
		Name	Size	Update date	
	okm:root	<input type="checkbox"/> 	CI0477978849.pdf	244.2 KB	25-01-2013 10:
	dicom	<input type="checkbox"/> 	CI0484641433.pdf	270.3 KB	25-01-2013 10:
	facturas	<input type="checkbox"/> 	CI0491320030.pdf	276.4 KB	25-01-2013 10:
	Scans	<input type="checkbox"/> 	CI0497971496.pdf	279.0 KB	25-01-2013 10:
		<input type="checkbox"/> 	CI0504591121.pdf	278.9 KB	25-01-2013 10:
		<input type="checkbox"/> 	CI0517932793.pdf	305.1 KB	25-01-2013 10:

Crontab sample - Document importer with metadata values

The script imports files with metadata values from the file system of the server.



- Files stored in the OpenKM server have a file name format like `some_text - (CUPS,REC).pdf`
- The variable **grpName** is the metadata group to be inserted.
- The variable **contractUUID** is the UUID of the OpenKM where the files will be imported.
- The variable **systemFolder** is the file system path.
- Verifies paths and extracts metadata from text between the characters "(" ")" present in the file name.
- Files are stored at `basePath/year/month` (when year or month folders are not present are automatically created).
- At the end, it send a mail with import results.

Metadata group definition:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.1//EN"
"http://www.openkm.com/dtd/property-groups-2.1.dtd">
<property-groups>
  <property-group label="Contract" name="okg:contract">
    <input label="Cups" type="text" name="okp:contract.cups" width="200px"/>
    <select label="Año" name="okp:contract.year" type="simple">
      <option label="2012" value="2012"/>
      <option label="2011" value="2011"/>
      <option label="2010" value="2010"/>
      <option label="2009" value="2009"/>
      <option label="2008" value="2008"/>
    </select>
    <select label="Mes" name="okp:contract.month" type="simple">
      <option label="Enero" value="enero"/>
      <option label="Febrero" value="febrero"/>
      <option label="Marzo" value="marzo"/>
      <option label="Abril" value="abril"/>
      <option label="Mayo" value="mayo"/>
      <option label="Junio" value="junio"/>
      <option label="Julio" value="julio"/>
      <option label="Agosto" value="agosto"/>
      <option label="Septiembre" value="septiembre"/>
      <option label="Octubre" value="octubre"/>
      <option label="Noviembre" value="noviembre"/>
      <option label="Diciembre" value="diciembre"/>
    </select>
  </property-group>
</property-groups>
```

The script:

```
import java.io.*;
import java.util.*;
import com.openkm.bean.Document;
```

```

import com.openkm.module.db.DbDocumentModule;
import com.openkm.dao.DatabaseMetadataDAO;
import com.openkm.util.DatabaseMetadataUtils;
import com.openkm.dao.bean.DatabaseMetadataValue;
import com.openkm.api.OKMRepository;
import com.openkm.api.OKMFolder;
import com.openkm.bean.Folder;
import com.google.gson.Gson;
import com.openkm.dao.NodeBaseDAO;
import java.lang.StringBuffer;
import com.openkm.core.*;
import java.net.URLEncoder;
import com.openkm.util.MailUtils;
import java.util.Date;
import com.openkm.module.db.stuff.DbSessionManager;
import com.openkm.extension.core.*;
import com.openkm.api.OKMDocument;
import com.openkm.api.OKMPropertyGroup;

class Status {
    public String fileName = "";
    public String dstPath = "";
    public String cups = "";
    public String month = "";
    public String year = "";
    public String error = "";
}

// Months
List month = new ArrayList();
month.add("ENERO");
month.add("FEBRERO");
month.add("MARZO");
month.add("ABRIL");
month.add("MAYO");
month.add("JUNIO");
month.add("JULIO");
month.add("AGOSTO");
month.add("SEPTIEMBRE");
month.add("OCTUBRE");
month.add("NOVIEMBRE");
month.add("DICIEMBRE");

String token = DbSessionManager.getInstance().getSystemToken();
String grpName = "okg:contract";
String contractUUID = "56aabd01-eeeb-47b3-bb10-86ee609541cf";
String systemFolder = "/home/openkm/pending_to_import_folder";
String basePath = OKMRepository.getInstance().getNodePath(token, contractUUID);
Gson gson = new Gson();
List bad = new ArrayList();
List good = new ArrayList();
List toAddress = new ArrayList();

// Loading files
File folder = new File(systemFolder);
File[] listOfFiles = folder.listFiles();

for (int i = 0; i < listOfFiles.length; i++) {
    File file = listOfFiles[i];
    if (file.isFile() && file.getName().toLowerCase().endsWith(".pdf")) {
        Status status = new Status();
        fileName = file.getName();
        status.fileName = fileName;

        if (fileName.indexOf("(") > 0 && fileName.indexOf("(") < fileName.indexOf(")")
            // Get metadata
            String metadata = fileName.substring((fileName.indexOf("(")+1), fileName.

```

```

// get data
String[] data = metadata.split(",");
if (data.length != 2) {
    status.error = "Incorrect format -> name (cups,REN)<br>";
} else {
    status.cups = data[0].toUpperCase().replaceAll(" ", "");
    Date date = new Date(file.lastModified());
    status.year = String.valueOf(1900 + date.getYear());
    status.month = month.get(date.getMonth());

    if (!data[1].toUpperCase().equals("REN")) {
        status.error = "Error type REN not found<br>";
    }

    if (status.error.equals("")) {
        // test if folder year exists otherside create it
        path = basePath + "/" + status.year;

        if (!OKMRepository.getInstance().hasNode(token, path)) {
            Folder folder = new Folder();
            folder.setPath(path);
            OKMFolder.getInstance().create(token, folder);
        }

        // test if folder month exists otherside create it
        path = path + "/" + status.month;

        if (!OKMRepository.getInstance().hasNode(token, path)) {
            Folder folder = new Folder();
            folder.setPath(path);
            OKMFolder.getInstance().create(token, folder);
        }

        // Create document and adding metadata
        try {
            // Removing extra ( contents )
            fileName = fileName.substring(0, fileName.indexOf("("));

            // Removing - and spaces at ends
            while (fileName.substring(fileName.length()-1).equals(" ") ||
                fileName.substring(0, fileName.length()-1).equals(" "))
                fileName = fileName.substring(0, fileName.length()-1);

            fileName = fileName + ".pdf";
            Document doc = new Document();
            path = path + "/" + fileName;
            doc.setPath(path);
            FileInputStream fis = new FileInputStream(file);
            doc = OKMDocument.getInstance().create(token, doc, fis);
            status.dstPath = path;

            // Create new metadata
            OKMPropertyGroup.getInstance().addGroup(token, doc.getPath(),
                Map properties = new HashMap();
            properties.put("okp:contract.cups", status.cups);
            properties.put("okp:contract.year", gson.toJson(new String[] {
            properties.put("okp:contract.month", gson.toJson(new String[] {
            OKMPropertyGroup.getInstance().setPropertiesSimple(token, doc

            // Delete file
            boolean success = file.delete();
            if (!success) {
                status.error += "File can not been deleted";
            }
        } catch (PathNotFoundException e) {
            status.error += "PathNotFoundException";

```

```

        } catch (ItemExistsException e) {
            status.error += "ItemExistsException";
        } catch (UnsupportedMimeTypeException e) {
            status.error += "UnsupportedMimeTypeException";
        } catch (FileSizeExceededException e) {
            status.error += "FileSizeExceededException";
        } catch (VirusDetectedException e) {
            status.error += "VirusDetectedException";
        } catch (RepositoryException e) {
            status.error += "RepositoryException";
        } catch (DatabaseException e) {
            status.error += "DatabaseException";
        } catch (ExtensionException e) {
            status.error += "ExtensionException";
        } catch (IOException e) {
            status.error += "IOException";
        } catch (Exception e) {
            status.error += e.getMessage();
        }
    }
} else {
    status.error = "Document format incorrect -> nombre - (cups,REN)<br>";
}

if (!status.error.equals("")) {
    bad.add(status);
} else {
    good.add(status);
}
}

StringBuffer result = new StringBuffer();
result.append("<h1>Import report</h1>");
result.append("<br><br>");
result.append("<table border=\"0\" cellpadding=\"2\" cellspacing=\"0\" width=\"100%\">");
result.append("<tr>");
result.append("<td bgcolor=\"silver\"><b>Name</b></td>");
result.append("<td bgcolor=\"silver\"><b>CUPS</b></td>");
result.append("<td bgcolor=\"silver\"><b>Año</b></td>");
result.append("<td bgcolor=\"silver\"><b>Mes</b></td>");
result.append("<td bgcolor=\"silver\"><b>Error</b></td>");
result.append("</tr>");
result.append("<tr>");
result.append("<td colspan=\"6\" bgcolor=\"silver\"><b>Errors:</b>"+bad.size()+"</b></td>");
result.append("</tr>");

for (Status status : bad) {
    result.append("<tr>");
    result.append("<td>"+status.fileName+"</td>");
    result.append("<td>"+status.cups+"</td>");
    result.append("<td>"+status.year+"</td>");
    result.append("<td>"+status.month+"</td>");
    result.append("<td><font color=\"red\">"+status.error+"</font></td>");
    result.append("</tr>");
}

result.append("</table>");
result.append("<br><br>");
result.append("<table border=\"0\" cellpadding=\"2\" cellspacing=\"0\" width=\"100%\">");
result.append("<tr>");
result.append("<td bgcolor=\"silver\"><b>Name</b></td>");
result.append("<td bgcolor=\"silver\"><b>CUPS</b></td>");
result.append("<td bgcolor=\"silver\"><b>Year</b></td>");
result.append("<td bgcolor=\"silver\"><b>Month</b></td>");
result.append("<td bgcolor=\"silver\"><b>Destination</b></td>");

```

```

result.append("</tr>");
result.append("<tr>");
result.append("<td colspan=\"6\" bgcolor=\"silver\"><b>Imported correctly:\""+good.size
result.append("</tr>");

for (Status status : good) {
    result.append("<tr>");
    result.append("<td>"+status.fileName+"</td>");
    result.append("<td>"+status.cups+"</td>");
    result.append("<td>"+status.year+"</td>");
    result.append("<td>"+status.month+"</td>");
    result.append("<td><a href=\""+Config.APPLICATION_URL+""?docPath="+URLLEncoder.encode
    result.append("</tr>");
}

result.append("</table>");

// Sending mails
toAddress.add("some@mail.com");
MailUtils.sendMessage(toAddress, "Importing report",result.toString());

print(result.toString());

```

Images









Register metadata definition

Group label		Group name			
Contract		okg:contract			
Label	Name	Width	Height	Field	
Cups	okp:contract.cups	200px	25px	Input	Readonly: false Data: Type: text
Año	okp:contract.year	150px	25px	Select	Readonly: false Data: Type: simple Options: <ul style="list-style-type: none">Label: 2012, VaLabel: 2011, VaLabel: 2010, VaLabel: 2009, VaLabel: 2008, Va
Mes	okp:contract.month	150px	25px	Select	Readonly: false Data: Type: simple Options: <ul style="list-style-type: none">Label: Enero, VLabel: Febrero,Label: Marzo, VLabel: Abril, VaLabel: Mayo, VLabel: Junio, VLabel: Julio, VaLabel: Agosto,Label: SeptiemLabel: Octubre,Label: NoviemtLabel: Diciemb

Create crontab task

Crontab					
Name	Expression	Mime	File Name	Mail	Last E
DashboardActivityPurge	@daily	application/x-bsh	DashboardActivityPurge.bsh	noreply@openkm.com	06-ene-20 0:00:00
Dashboard Activity Purge	@daily	application/x-bsh	DashboardActivityPurge.bsh	noreply@openkm.com	06-ene-20 0:00:00
User Items Serialize	@hourly	application/x-bsh	UserItemsSerialize.bsh	noreply@openkm.com	06-ene-20 10:00:00
User Mail Importer	*/30 * * * *	application/x-bsh	UserMailImporter.bsh	noreply@openkm.com	06-ene-20 10:30:00
Text Extractor Worker	*/5 * * * *	application/x-bsh	TextExtractorWorker.bsh	noreply@openkm.com	06-ene-20 10:45:00
Repository Info	@daily	application/x-bsh	RepositoryInfo.bsh	noreply@openkm.com	06-ene-20 0:00:00
Session Watchdog	*/5 * * * *	application/x-bsh	SessionWatchdog.bsh	noreply@openkm.com	06-ene-20 10:45:00
Process Pending Tasks	*/5 * * * *	application/x-bsh	ProcessPendingTasks.bsh	noreply@openkm.com	06-ene-20 10:45:00
Import server data	0 6 * * *	application/x-bsh	import.bsh		

Files in application server









 document 1 - (CUPS001025,REN).pdf	5,9 KB Adobe Acrobat Document
 document 2 - (CUPS001026,REN).pdf	5,9 KB Adobe Acrobat Document
 document 3 - (CUPS001046,REN).pdf	5,9 KB Adobe Acrobat Document
 document 4 - (CUPS001030,REN).pdf	5,9 KB Adobe Acrobat Document
 document 5 - (CUPS001045,REN).pdf	5,9 KB Adobe Acrobat Document
 document 6 - (CUPS001026).pdf	5,9 KB Adobe Acrobat Document
 document 7 - (CUPS001046,REN).pdf	5,9 KB Adobe Acrobat Document
 document 8 - CUPS001030 REC.pdf	5,9 KB Adobe Acrobat Document

Result**Mail notification**

Import report

Name	CUPS	Año	Mes	Error
Errors:2				
document 6 - (CUPS001026).pdf				Incorrect format
document 8 - CUPS001030 REC.pdf				Document format
Name	CUPS	Year	Month	De
Imported correctly:6				
document 2 - (CUPS001026,REN).pdf	CUPS001026	2012	NOVIEMBRE	/o
document 4 - (CUPS001030,REN).pdf	CUPS001030	2012	NOVIEMBRE	/o
document 3 - (CUPS001046,REN).pdf	CUPS001046	2012	NOVIEMBRE	/o
document 7 - (CUPS001046,REN).pdf	CUPS001046	2012	NOVIEMBRE	/o
document 5 - (CUPS001045,REN).pdf	CUPS001045	2012	NOVIEMBRE	/o
document 1 - (CUPS001025,REN).pdf	CUPS001025	2012	NOVIEMBRE	/o

Imported files into the repository

 document 1 - (CUPS001025,REN).pdf	5,9 KB Adobe Acrobat Document
 document 2 - (CUPS001026,REN).pdf	5,9 KB Adobe Acrobat Document
 document 3 - (CUPS001046,REN).pdf	5,9 KB Adobe Acrobat Document
 document 4 - (CUPS001030,REN).pdf	5,9 KB Adobe Acrobat Document
 document 5 - (CUPS001045,REN).pdf	5,9 KB Adobe Acrobat Document
 document 6 - (CUPS001026).pdf	5,9 KB Adobe Acrobat Document
 document 7 - (CUPS001046,REN).pdf	5,9 KB Adobe Acrobat Document
 document 8 - CUPS001030 REC.pdf	5,9 KB Adobe Acrobat Document

Script - Correct security in personal nodes

The script changes the security recursively for all the childs for the nodes **okm:personal**, **okm:trash** and **okm:mail** removing all previous security an assigning all grants to the owner of the folder (the folder name identifies the user).

```
import com.openkm.api.OKMAuth;
import com.openkm.api.OKMFolder;
import com.openkm.bean.Folder;
import java.util.*;
import com.openkm.util.PathUtils;
import com.openkm.bean.Permission;

public void changeSecurity(String path){
    for(Folder fld : OKMFolder.getInstance().getChildren(null, path)) {
        String userId = PathUtils.getName(fld.getPath());

        Map revokeRoles = OKMAuth.getInstance().getGrantedRoles(null, fld.getPath());
        Map revokeUsers = OKMAuth.getInstance().getGrantedUsers(null, fld.getPath());
        Map grantUsers = new HashMap();
        grantUsers.put(userId, Permission.ALL_GRANTS);
        revokeUsers.remove(userId);
        OKMAuth.getInstance().changeSecurity(null, fld.getPath(), grantUsers, revokeUsers);

        print("Done " + path + "<br />");
    }
}

changeSecurity("/okm:personal");
changeSecurity("/okm:trash");
changeSecurity("/okm:mail");
```

PHP SOAP Webservices samples



Webservices based on SOAP will be no longer supported. We encourage youu sing [SDK for PHP](#) that builds a complete client based on REST Webservices.

Script	Description
PHP SOAP Webservices samples - Automatic import	The script import pdf files from a folder in the file system of the application server.
PHP SOAP Webservices samples - Download a file	Download a file.
PHP SOAP Webservices samples - File upload	Upload a file.

PHP SOAP Webservices samples - Automatic import

Import the PDF files from some local folder **/converted/*.pdf** path to a specific OpenKM server path.

```
<?php
// Register WSDL
$OKMAuth = new SoapClient('http://localhost:8080/OpenKM/services/OKMAuth?wsdl');
$OKMDocument = new SoapClient('http://localhost:8080/OpenKM/services/OKMDocument?wsdl');
$OKMFolder = new SoapClient('http://localhost:8080/OpenKM/services/OKMFolder?wsdl');
// Login
$loginResp = $OKMAuth->login(array('user' => 'okmAdmin', 'password' => 'admin'));
$token = $loginResp->return;

foreach (glob("converted/*.pdf") as $filename) {
    echo "$filename size " . filesize($filename) . "<br />";

    // open file and encode if necessary
    $handle = fopen($filename, 'rb');
    $file_content = fread($handle, filesize($filename));
    fclose($handle);
    $encoded = base64_encode($file_content);
    $filename1 = substr($filename, 12);
    echo "$filename1" . "<br />";
    $OKMDocument->createSimple(array('token' => $token, 'docPath' => "/okm:root/import/"));

    //delete the file uoloaded
    unlink("c:/converted/".$filename1);
}

// Logout
$OKMAuth->logout($token);
?>
```

PHP SOAP Webservices samples - Download a file

Download a file.



The content type of the header is always set to **Content-type: application/pdf**

```
<?php
// Register WSDL
$OKMAuth = new SoapClient('http://localhost:8080/OpenKM/services/OKMAuth?wsdl');
$OKMDocument = new SoapClient('http://localhost:8080/OpenKM/services/OKMDocument?wsdl');
$OKMFolder = new SoapClient('http://localhost:8080/OpenKM/services/OKMFolder?wsdl');

// Login
$loginResp = $OKMAuth->login(array('user' => 'okmAdmin', 'password' => 'admin'));
$token = $loginResp->return;

// Open document test
$docPath = '/okm:root/Customer Docs/foo/Invoice-0000034.pdf';

// Get document content
$getDocumentResp = $OKMDocument->getContent(array('token' => $token, 'docId' => $docPath));
$length = strlen($getDocumentResp->return);

// Display document response as pdf
header("Content-type: application/pdf");
header("Content-Length: $length");
header("Content-Disposition: inline; filename='test.pdf'");
echo $getDocumentResp->return;

// Logout
$OKMAuth->logout(array('token' => $token));
?>
```

PHP SOAP Webservices samples - File upload

```
<?php
    $impfile = $argv[1];

    // Register WSDL
    $OKMAuth = new SoapClient('http://localhost:8080/OpenKM/services/OKMAuth?wsdl');
    $OKMDocument = new SoapClient('http://localhost:8080/OpenKM/services/OKMDocument?wsdl');
    // Login
    $loginResp = $OKMAuth->login(array('user' => 'okmAdmin', 'password' => 'admin'));
    $token = $loginResp->return;

    // open file and encode if necessary
    $handle = fopen($impfile,'rb');
    $file_content = fread($handle,filesize($impfile));
    fclose($handle);
    $encoded = base64_encode($file_content);

    // Login
    $OKMDocument->createSimple(array('token' => $token, 'docPath' => "/okm:root/file.to"));

    // Logout
    $OKMAuth->logout($token);
?>
```

Configurable shell script that uses PHP support to import files.

```
#!/bin/bash
# JOAKO Import 1.0.0 - Copyright 2011 Andrew Joakimsen
# Distributed under the terms of the GNU General Public
# License Version 2 and no future versions.
#
# Script to import files into OpenKM and report error/success.
# Requires the PHP import script written by snowman, with a
# few small tweaks.

#####
###                               ###
###  C O N F I G U R A T I O N  ###
###                               ###
#####

# OpenKM directory where files are imported to
OKM_IMPORT_DIR=NEW

# Filesystem directory (no trailing slash) where we put imported files (it will be on
STORE_DIR=~/.we-test

# Filesystem directory (no trailing slash) where we put files that encountered import
ERROR_STORE_DIR=~/.error-test

#####
# First let's see if the runtime condition is sane
if [ -z "$1" ]; then
    echo "JOAKO Import: Missing file name"
    echo "Usage: import.sh [filename] [target-directory]"
    exit 1
fi

# If ARG1 is a real file
if [ -d "$1" ]; then
```

```

        echo "ERROR: \"$1\" is a directory!"
        exit 1
    fi
    if [ ! -f "$1" ]; then
        echo "ERROR: \"$1\" Invalid file name "
        exit 1
    else
        INFILE=$1
    fi

    # Check if OKM_IMPORT_DIR was passed as argument
    if [ -z "$2" ]; then
        OKM_DIR=$OKM_IMPORT_DIR
    else
        OKM_DIR=$2
    fi

    # Check if the imported files directory exists
    if [ ! -d $STORE_DIR/$OKM_DIR ]; then
        if [ ! -d ~/$STORE_DIR ]; then
            mkdir $STORE_DIR
        fi
        mkdir $STORE_DIR/$OKM_DIR
    fi

    # Run the import Process
    RESULT=$(php /opt/openkm/import.php $INFILE ${INFILE##*/} $OKM_DIR 2>&1)
    RESULTCODE=$?

    # Deal with the consequences
    function importer-error {
        # Check if the import error directory exists
        if [ ! -d $ERROR_STORE_DIR/$OKM_DIR ]; then
            if [ ! -d $ERROR_STORE_DIR ]; then
                mkdir $ERROR_STORE_DIR
            fi
            mkdir $ERROR_STORE_DIR/$OKM_DIR
        fi
        echo "`date` JOAKO Import Encountered an error type $RESULTCODE."
        echo "$RESULT"
        mv $INFILE $ERROR_STORE_DIR/$OKM_DIR &> /dev/null
        exit $?
    }

    if [ $RESULTCODE -ne 0 ]; then # There was an error
        importer-error
    else # We check further
        if [ -n "$RESULT" ]; then # There was an error
            importer-error
        else # No error detected
            echo "`date` JOAKO Import: Imported file [${INFILE##*/}] to [$OKM_DIR] direct
            mv $INFILE $STORE_DIR/$OKM_DIR &> /dev/null
            exit 0
        fi
    fi

```

Crontab task - XML importer

Import files with metadata values from XML files.



This is a crontab task base in Java class packaged in a JAR.

- The files to be imported are stored at `/home/openkm/import/Output` OpenKM server. (Variable **SYSTEM_FOLDER**+Output).
- The XML files with metadata values are stored at `/home/openkm/import/logfile` OpenKM server. (Variable **SYSTEM_FOLDER**+logfile).
- The files are imported at OpenKM folder `"/okm:root/import"`. (Variable **OPENKM_FOLDER**).

The crontab does three actions:

- First import all files from `"/home/openkm/import/Output"` and delete them from server, after has being imported.
- Then load and parse xml files located at `"/home/openkm/import/logfile"` and set the metadata group values of the documents.
- Finally the xml files are deleted.

Metada group definition:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.1//EN"
    "http://www.openkm.com/dtd/property-groups-2.1.dtd">
<property-groups>
  <property-group label="Data" name="okg:data">
    <input label="Folio" name="okp:data.folio" type="text"/>
    <input label="Rut" name="okp:data.rut" type="text"/>
    <input label="Placa patente" name="okp:data.placa" type="text"/>
    <input label="Nombre" name="okp:data.nombres" type="text"/>
    <input label="Apellido parterno" name="okp:data.apellido_padre" type="text"/>
    <input label="Apellido materno" name="okp:data.apellido_madre" type="text"/>
    <select label="Año contable" name="okp:data.year" type="simple">
      <option value="2012" label="2012" />
      <option value="2013" label="2013" />
    </select>
    <input label="Tipo documento" name="okp:data.documento" type="text"/>
  </property-group>
</property-groups>
```

XML files structure:

```
<?xml version="1.0" encoding="UTF-16"?>
<INDEX_LOG>
  <Batch BatchID="2013-04-16">
    <File>
      <Filename>10201200365212 - 01.pdf</Filename>
```



```

        <Fields>
            <Field>
                <Name>Folio</Name>
                <Value>10201200365245</Value>
            </Field>
            <Field>
                <Name>Rut</Name>
                <Value>9419475-JK</Value>
            </Field>
            <Field>
                <Name>Placa Patente</Name>
                <Value>XG412190</Value>
            </Field>
            <Field>
                <Name>Nombres</Name>
                <Value>JOSEP</Value>
            </Field>
            <Field>
                <Name>Apellido Paterno</Name>
                <Value>LLORT</Value>
            </Field>
            <Field>
                <Name>Apellido Materno</Name>
                <Value>TELLA</Value>
            </Field>
            <Field>
                <Name>Año</Name>
                <Value>2012</Value>
            </Field>
            <Field>
                <Name>Tipo de Documento</Name>
                <Value>Comprobante de Permiso</Value>
            </Field>
        </Fields>
    </File>
    <!-- etc. -->
</Batch>
</INDEX_LOG>

```

JAVA class:

```

package com.openkm.crontab;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.util.Collection;
import java.util.HashMap;
import java.util.Map;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

import com.openkm.api.OKMDocument;
import com.openkm.api.OKMPropertyGroup;
import com.openkm.api.OKMSearch;
import com.openkm.automation.AutomationException;
import com.openkm.bean.Document;

```

```

import com.openkm.bean.PropertyGroup;
import com.openkm.bean.QueryResult;
import com.openkm.core.AccessDeniedException;
import com.openkm.core.DatabaseException;
import com.openkm.core.FileSizeExceededException;
import com.openkm.core.ItemExistsException;
import com.openkm.core.LockException;
import com.openkm.core.NoSuchGroupException;
import com.openkm.core.NoSuchPropertyException;
import com.openkm.core.ParseException;
import com.openkm.core.PathNotFoundException;
import com.openkm.core.RepositoryException;
import com.openkm.core.UnsupportedMimeTypeException;
import com.openkm.core.UserQuotaExceededException;
import com.openkm.core.VirusDetectedException;
import com.openkm.dao.bean.QueryParams;
import com.openkm.extension.core.ExtensionException;
import com.openkm.module.db.stuff.DbSessionManager;

/**
 * XMLImporter
 */
public class XMLImporter {
    private static final String SYSTEM_FOLDER = "/home/openkm/import";
    private static final String OPENKM_FOLDER = "/okm:root/import";
    private static final String ATTRIBUTE_FILENAME = "Filename";
    private static final String ATTRIBUTE_FIELDS = "Fields";
    private static final String ATTRIBUTE_FIELD = "Field";
    private static final String ATTRIBUTE_NAME = "Name";
    private static final String ATTRIBUTE_VALUE = "Value";
    private static final String FIELD_NAME_FOLIO = "folio";
    private static final String FIELD_NAME_RUT = "rut";
    private static final String FIELD_NAME_PLACA_PATENTE = "placa patente";
    private static final String FIELD_NAME_NOMBRES = "nombres";
    private static final String FIELD_NAME_APELLIDO_PATERNO = "apellido paterno";
    private static final String FIELD_NAME_APELLIDO_MATERNO = "apellido materno";
    private static final String FIELD_NAME_AÑO = "año";
    private static final String FIELD_NAME_TIPO_DOCUMENTO = "tipo de documento";

    public static void main(String[] args) {
        cronTask(DbSessionManager.getInstance().getSystemToken());
    }

    public static String cronTask(String systemToken) {
        try {
            importFiles();
            importMetadata();
        } catch (Exception e) {
            e.printStackTrace();
        }

        return "";
    }

    /**
     * importMetadata
     */
    public static void importMetadata() throws ParserConfigurationException, SAXException,
        RepositoryException, DatabaseException, PathNotFoundException, NoSuchGroupException,
        ExtensionException, NoSuchPropertyException, UnsupportedMimeTypeException,
        UserQuotaExceededException, VirusDetectedException, ItemExistsException,
        String fileName = "";
        String folio = "";
        String rut = "";
        String placaPatente = "";
        String nombres = "";

```

```

String apellidoPaterno = "";
String apellidoMaterno = "";
String ano = "";
String tipoDocumento = "";

File folder = new File(SYSTEM_FOLDER + "/logfile");
File[] listOfFiles = folder.listFiles();

for (int i = 0; i < listOfFiles.length; i++) {
    File xmlFile = listOfFiles[i];
    if (xmlFile.isFile() && xmlFile.getName().toLowerCase().endsWith(".ind")) {
        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
        dbf.setNamespaceAware(true);
        dbf.setAttribute("http://xml.org/sax/features/namespaces", Boolean.FALSE);

        DocumentBuilder db = dbf.newDocumentBuilder();
        org.w3c.dom.Document xmlDoc = db.parse(xmlFile);
        xmlDoc.getDocumentElement().normalize();
        NodeList fileNodesList = xmlDoc.getElementsByTagName("File");

        for (int x = 0; x < fileNodesList.getLength(); x++) {
            fileName = "";
            folio = "";
            rut = "";
            placaPatente = "";
            nombres = "";
            apellidoPaterno = "";
            apellidoMaterno = "";
            ano = "";
            tipoDocumento = "";
            Node childFildNode = fileNodesList.item(x);

            if (childFildNode.getNodeType() == Node.ELEMENT_NODE) {
                Element fileElement = (Element) childFildNode; // attribute
                fileName = fileElement.getElementsByTagName(ATTRIBUTE_FILENAME).item(0).getTextContent();
                Node fieldsNode = fileElement.getElementsByTagName(ATTRIBUTE_FIELDS).item(0);

                // Fields
                if (fieldsNode.getNodeType() == Node.ELEMENT_NODE) {
                    Element fieldsElement = (Element) fieldsNode; // fields
                    // node
                    NodeList fieldList = fieldsElement.getElementsByTagName(ATTRIBUTE_FIELDS);
                    for (int y = 0; y < fieldList.getLength(); y++) {
                        Node fieldNode = fieldList.item(y);
                        if (fieldNode.getNodeType() == Node.ELEMENT_NODE) {
                            Element fieldElement = (Element) fieldNode; // field
                            // node
                            String name = fieldElement.getElementsByTagName(ATTRIBUTE_FIELDS).item(0).getTextContent();
                            String value = fieldElement.getElementsByTagName(ATTRIBUTE_FIELDS).item(1).getTextContent();

                            if (name.toLowerCase().equals(FIELD_NAME_FOLIO)) {
                                folio = value;
                            } else if (name.toLowerCase().equals(FIELD_NAME_RUT)) {
                                rut = value;
                            } else if (name.toLowerCase().equals(FIELD_NAME_PLACA_PATENTE)) {
                                placaPatente = value;
                            } else if (name.toLowerCase().equals(FIELD_NAME_NOMBRES)) {
                                nombres = value;
                            } else if (name.toLowerCase().equals(FIELD_NAME_APELLIDO_PATERNO)) {
                                apellidoPaterno = value;
                            } else if (name.toLowerCase().equals(FIELD_NAME_APELLIDO_MATERNO)) {
                                apellidoMaterno = value;
                            } else if (name.toLowerCase().equals(FIELD_NAME_ANO)) {
                                ano = value;
                            } else if (name.toLowerCase().equals(FIELD_NAME_TIPO_DOCUMENTO)) {
                                tipoDocumento = value;
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

                                tipoDocumento = value;
                            }
                        }
                    }

                    if (fileName != null && !fileName.equals("")) {
                        QueryParams queryParams = new QueryParams();
                        queryParams.setDomain(QueryParams.DOCUMENT);
                        queryParams.setName(fileName);
                        Collection<QueryResult> results = OKMSearch.getInstance().search(queryParams);

                        if (results.size() == 1) {
                            for (QueryResult queryResult : results) {
                                if (queryResult.getDocument() != null) {
                                    System.out.println(fileName + " -ok");
                                    boolean found = false;

                                    for (PropertyGroup group : OKMPropertyGroup.getInstance().getGroups()) {
                                        if (group.getName().equals("okg:data")) {
                                            found = true;
                                        }
                                    }

                                    if (!found) {
                                        OKMPropertyGroup.getInstance().addGroup("okg:data");
                                    }

                                    Map<String, String> propertiesMap = new HashMap<>();
                                    propertiesMap.put("okp:data.folio", folio);
                                    propertiesMap.put("okp:data.rut", rut);
                                    propertiesMap.put("okp:data.placa", placa);
                                    propertiesMap.put("okp:data.nombres", nombres);
                                    propertiesMap.put("okp:data.apellido_paterno", apellidoPaterno);
                                    propertiesMap.put("okp:data.apellido_materno", apellidoMaterno);
                                    propertiesMap.put("okp:data.year", ano);
                                    propertiesMap.put("okp:data.documento", tipoDocumento);
                                    OKMPropertyGroup.getInstance().setProperty("okp:data", propertiesMap);
                                }
                            }
                        } else if (results.size() > 1) {
                            System.out.println(fileName + " - error");
                        } else {
                            System.out.println(fileName + " - not found");
                        }
                    }
                }
            }

            xmlFile.delete();
        }
    }

    /**
     * importFiles
     */
    public static void importFiles() throws UnsupportedOperationException, FileNotFoundException, VirusDetectedException, ItemExistsException, PathNotFoundException, AccessDeniedException, DatabaseException, ExtensionException, AutomationException {
        String systemToken = DbSessionManager.getInstance().getSystemToken();
        // Loading files
        File folder = new File(SYSTEM_FOLDER + "/Output");
        File[] listOfFiles = folder.listFiles();
        for (int i = 0; i < listOfFiles.length; i++) {
            File file = listOfFiles[i];
            if (file.isFile() && file.getName().toLowerCase().endsWith(".pdf")) {

```

```

        Document doc = new Document();
        doc.setPath(OPENKM_FOLDER + "/" + file.getName());
        FileInputStream fis = new FileInputStream(file);
        doc = OKMDocument.getInstance().create(systemToken, doc, fis);
        file.delete();
    }
}
}

```

Images

Create the jar file



More information at [Creating a JAR file with Eclipse](#).




Register crontab task

Crontab					
Name	Expression	Mime	File Name	Mail	Last E
DashboardActivityPurge	@daily	application/x-bsh	DashboardActivityPurge.bsh	noreply@openkm.com	06-ene-20 0:00:00
Dashboard Activity Purge	@daily	application/x-bsh	DashboardActivityPurge.bsh	noreply@openkm.com	06-ene-20 0:00:00
User Items Serialize	@hourly	application/x-bsh	UserItemsSerialize.bsh	noreply@openkm.com	06-ene-20 10:00:00
User Mail Importer	*/30 * * * *	application/x-bsh	UserMailImporter.bsh	noreply@openkm.com	06-ene-20 10:30:00
Text Extractor Worker	*/5 * * * *	application/x-bsh	TextExtractorWorker.bsh	noreply@openkm.com	06-ene-20 10:45:00
Repository Info	@daily	application/x-bsh	RepositoryInfo.bsh	noreply@openkm.com	06-ene-20 0:00:00
Session Watchdog	*/5 * * * *	application/x-bsh	SessionWatchdog.bsh	noreply@openkm.com	06-ene-20 10:45:00
Process Pending Tasks	*/5 * * * *	application/x-bsh	ProcessPendingTasks.bsh	noreply@openkm.com	06-ene-20 10:45:00
Import server data	0 6 * * *	application/x-bsh	import.bsh		

Register metadata definition

Group label		Group name			
Contract		okg:contract			
Label	Name	Width	Height	Field	
Cups	okp:contract.cups	200px	25px	Input	Readonly: false Data: Type: text
Año	okp:contract.year	150px	25px	Select	Readonly: false Data: Type: simple Options: <ul style="list-style-type: none">Label: 2012, VaLabel: 2011, VaLabel: 2010, VaLabel: 2009, VaLabel: 2008, Va
Mes	okp:contract.month	150px	25px	Select	Readonly: false Data: Type: simple Options: <ul style="list-style-type: none">Label: Enero, VLabel: Febrero,Label: Marzo, VLabel: Abril, VaLabel: Mayo, VLabel: Junio, VLabel: Julio, VaLabel: Agosto,Label: SeptiemLabel: Octubre,Label: NoviemtLabel: Diciemb

Logfile folder

	home	openkm	logfile
	2013-04-16 - IndexLog.xml	517,2 KB documento XML	
	2013-04-15 - IndexLog.xml	851,5 KB documento XML	

Output folder

	home	openkm	Output
	12201200359104 - 01.pdf	51,2 KB	Adobe Acrobat Document
	12201200359081 - 01.pdf	53,5 KB	Adobe Acrobat Document
	12201200359044 - 04.pdf	415,1 KB	Adobe Acrobat Document
	12201200359044 - 03.pdf	31,6 KB	Adobe Acrobat Document
	12201200359044 - 02.pdf	39,3 KB	Adobe Acrobat Document
	12201200359044 - 01.pdf	58,8 KB	Adobe Acrobat Document

Result

The screenshot displays the OpenKM web interface. On the left, the 'Taxonomy' sidebar shows a tree structure with 'okm:root' at the top, followed by 'facturas', 'import' (highlighted with a red box), 'mails', and 'records'. Below this, there are sections for 'Categories', 'Metadata', 'Templates', 'My documents', 'E-mail', and 'Trash'. The main content area shows a file list for the path '/okm:root/import/'. The list includes columns for 'Name', 'Title', 'Language', and 'Size'. The first file, '10201200365212 - 01.pdf', is highlighted with a red box. Below the file list, the 'Datos' (Data) tab is active, showing a table of metadata for the selected file, which is also highlighted with a red box. The status bar at the bottom indicates 'Connected as Administrator' and shows various system metrics.

Name	Title	Language	Size
10201200365212 - 01.pdf		español	50.0 KB
10201200365212 - 02.pdf			41.5 KB
10201200365212 - 03.pdf		portugués	45.4 KB
10201200365212 - 04.pdf			86.7 KB
10201200365212 - 05.pdf		estonio	30.0 KB
10201200365213 - 01.pdf		italiano	49.8 KB
10201200365213 - 02.pdf			40.8 KB
10201200365213 - 03.pdf		portugués	38.7 KB
10201200365213 - 04.pdf		español	22.5 KB

Folio	10201200365245
Rut	9419475-JK
Placa patente	XG412190
Nombre	JOSEP
Apellido parterno	LLORT
Apellido materno	TELLA
Año contable	2012
Tipo documento	Comprobante de Permiso

Vault of stuff

<i>Field / Property</i>	<i>Description</i>
<i><u>Convert Exchange Legacy DN</u></i>	<i>Changes all the Legacy DN strings into proper mail addresses.</i>
<i><u>Download document with direct link</u></i>	<i>Downloads a document by url.</i>
<i><u>Import document with bash script using REST</u></i>	<i>Imports a document with a bash script that uses REST Webservices API.</i>
<i><u>Import document with bash script using SOAP</u></i>	<i>Imports a document with a bash script that uses SOAP Webservices API.</i>
<i><u>Creating users from scripts</u></i>	<i>Creation of users from generated scripts with file.</i>

Convert Exchange Legacy DN

The script changes all the Legacy DN strings into proper mail addresses. Therefore there's an export file of the global address list from our IT as .csv containing all the Exchange Legacy DN's and their corresponding mail addresses. With this data the database can be altered the Exchange Legacy DN can be changed to normal mail addresses.

CSV sample:

```
/o=EMail/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=somefanc
/o=EMail/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=anotherf
/o=EMail/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=m.mueller
```

```
<?php
# Change ExchangeDN to Mailaddresses
# Malte Küppers
#
# CSV Legacy File Format:      Exchange Legacy DN ; Mailadresse
#
#

# variables
$row = 0;
$legacy_file = "/root/Hilfsdateien/LegacyExchangeDN.txt" # Set to your filepath

if (($handle = fopen($legacy_file, "r")) !== FALSE) {
    while (($data = fgetcsv($handle, 1000, ";")) !== FALSE) {
        $exchdn[$row]=strtoupper($data[0]);      #Exchange Legacy DN
        $mail[$row]=$data[1];                      #Mailadresse
        $row++;
    }
    fclose($handle);
    #echo $row ." lines read";
}

#Open Database Connection
$db = mysqli_connect("127.0.0.1", "<USERNAME>", "<PASSWORD>", "okmdb");
if(!$db)
{
    exit("Verbindungsfehler: ".mysqli_connect_error());
}

# Mail FROM ändern
$ergebnis = mysqli_query($db, "SELECT DISTINCT NML_FROM FROM OKM_NODE_MAIL");
while($row = mysqli_fetch_object($ergebnis)){
    #only adresses that needs to be changed (Starting with /o= )
    if(strpos($row->NML_FROM, "/o")!==false) {

        $key = array_search(strtoupper($row->NML_FROM), $exchdn);
        if(is_numeric($key)){
            echo $row->NML_FROM ." found at Pos #". $key . "<br>";
            $qry = "UPDATE OKM_NODE_MAIL SET NML_FROM = '". $mail[
            #echo $qry . "<br><br>";
            $update = mysqli_query($db, $qry);
        }
    }
}
```

```

}

# Mail TO ändern
$ergebnis = mysqli_query($db, "SELECT DISTINCT NML_TO FROM OKM_NODE_MAIL_TO");
while($row = mysqli_fetch_object($ergebnis)){
    #Nur Adressen die nicht geändert wurden
    if(strpos($row->NML_TO,"/o")!==false) {

        $key = array_search(strtoupper($row->NML_TO), $exchdn);
        if(is_numeric($key)){
            echo $row->NML_TO ." gefunden an Pos #".$key ."<br>";
            $qry = "UPDATE OKM_NODE_MAIL_TO SET NML_TO = '". $mail;
            #echo $qry . "<br><br>";
            $update = mysqli_query($db,$qry);
        }

    }

}

# Mail CC ändern
$ergebnis = mysqli_query($db, "SELECT DISTINCT NML_CC FROM OKM_NODE_MAIL_CC");
while($row = mysqli_fetch_object($ergebnis)){
    #Nur Adressen die nicht geändert wurden
    if(strpos($row->NML_CC,"/o")!==false){

        $key = array_search(strtoupper($row->NML_CC), $exchdn);
        if(is_numeric($key)){
            echo $row->NML_CC ." gefunden an Pos #".$key ."<br>";
            $qry = "UPDATE OKM_NODE_MAIL_CC SET NML_CC = '". $mail;
            #echo $qry . "<br><br>";
            $update = mysqli_query($db,$qry);
        }

    }

}

# Mail BCC ändern
$ergebnis = mysqli_query($db, "SELECT DISTINCT NML_BCC FROM OKM_NODE_MAIL_BCC");
while($row = mysqli_fetch_object($ergebnis)){
    #Nur Adressen die nicht geändert wurden
    if(strpos($row->NML_BCC,"/o")!==false){

        $key = array_search(strtoupper($row->NML_BCC), $exchdn);
        if(is_numeric($key)){
            echo $row->NML_BCC ." gefunden an Pos #".$key ."<br>";
            $qry = "UPDATE OKM_NODE_MAIL_BCC SET NML_BCC = '". $ma;
            #echo $qry . "<br><br>";
            $update = mysqli_query($db,$qry);
        }

    }

}

?>

```



Additional information at our public [forum](#).

Download document with direct link

You can download a document from OpenKM using a command line utility like **wget**. There is a service with BASIC authentication for this task. You can reference a document by UUID using the **uuid** parameter or by path, using the **path** parameter. In the same URL you need to specify the user and password.



When you share the user and password for downloading files is a good practice configure the user profile disallowing all the features. That will lock UI access to the user.

Available parameters:

- The parameter **uuid** is the document UUID
- The parameter **path** is the document path.
- The parameter **inline** set when document must be downloaded as attachment or shown inline. It's a boolean parameter, allowed values are true or false.



When you use parameter path, it is a good practice to encode url to prevent problems with special characters.

Sample:

```
http://user:password@localhost:8080/OpenKM/Download?uuid=4ca63b87-e340-443e-aeef-6c7e
http://user:password@localhost:8080/OpenKM/Download?path=/okm:root/test/test.doc
```

This ease some integration tasks:

```
$ wget http://okmAdmin:admin@localhost:8080/OpenKM/Download?uuid=xxx-yyy-zzz
```

Import document with bash script using REST

Simplified version of the script which make use of RESTful API.

```
#!/bin/bash
OKM_SERVICE="http://localhost:8080/OpenKM"
OKM_USER="okmAdmin"
OKM_PASSWORD="admin"
OKM_PATH="/okm:root/upload"
LOCAL_PATH="."
TMP=$(mktemp)
RESET='\e[0m'
ERROR='\e[1;31m'
OK='\e[1;34m'

for DIR in $(find $LOCAL_PATH -type d); do
    DIR=${DIR:1:${#DIR}}
    if [ -n "$DIR" ]; then
        RETCODE=$(curl -u $OKM_USER:$OKM_PASSWORD --write-out %{http_code} -X POST --silent \
            -H "Content-Type: application/json" -d "$OKM_PATH$DIR" "$OKM_SERVICE/services/rest/document")
        if [ $RETCODE != "200" ]; then
            echo -e "${ERROR}ERROR -> $(cat $TMP){RESET}";
        else
            echo -e "${OK}Folder created: ${OKM_PATH}$DIR{RESET}"
        fi
    fi
done

for FILE in $(find $LOCAL_PATH -type f); do
    FILE=${FILE:2:${#FILE}}
    RETCODE=$(curl -u $OKM_USER:$OKM_PASSWORD --write-out %{http_code} -X POST --silent \
        -F "docPath=$OKM_PATH/$FILE" -F content=@$FILE "$OKM_SERVICE/services/rest/document")
    if [ $RETCODE != "200" ]; then
        echo -e "${ERROR}ERROR -> $(cat $TMP){RESET}";
    else
        echo -e "${OK}Document created: ${OKM_PATH}/${FILE}{RESET}"
    fi
done

# Cleanup
rm -f $TMP
```

Import document with bash script using SOAP

The script imports a document into OpenKM server using SOAP Webservices.

Download the script [import_sh.zip](#).

```
#!/bin/sh
#####
# Copyright (c) 2013: Jörg Palmer, JoergPalmer @ OpenKM forums
#
# Script for importing a document into OpenKM via web services
# 1. Logon using an "import user"
# 2. Import the document into a pre-defined folder
# 3. Logoff the server
#####

### Configuration section, please adjust to your setup
OKM_ID="autoImport"           # User ID / name
OKM_PW="Import42"             # Password
### Configuration section ends here
### DO NOT CHANGE FROM HERE UNLESS YOU KNOW WHAT YOU ARE DOING!

# Constants definition
TOOLNAME="OpenKM_import"
VERSION="v1.0"

# Possible exit codes
EXIT_OK="0"                   # 0=success
EXIT_BAD_ARGS="1"             # 1=wrong number of parameters
EXIT_BAD_LOGON="2"             # 2=logon to OpenKM failed
EXIT_BAD_LOGOFF="3"           # 3=Logoff from OpenKM failed

# LOG Level
LOG_ERR="0"                   # 0=only error messages
LOG_INFO="1"                  # 1=error messages and some infos
LOG_DEBUG="2"                 # 2=debug level logging

START=`date +%s`

usage() {
    cat << EOF
-----
Script to import a document into an OpenKM server, using SOAP web services.
Please adjust the user id settings at the beginning of the script to your needs.

Copyright: Jörg Palmer
Version: $VERSION

Usage: OpenKM_import.sh [-h] [-v] [-g] [-u url] [-p path] document

Options:

-h      : Display this help message
-v      : Increase the verbosity (this option can be used more than once)
-g      : Activate debug mode:
          - Set the verbosity to the highest possible
-u url   : Set the url to the OpenKM instance
          Default: http://localhost:8080/OpenKM
-p path  : Set the taxonomy path to be imported to
          Default: /okm:root/imported
```

```

document : The file to be imported
-----
EOF
}

#####
# Get an absolute path from a relative path to a file
#
# Param1 : Relative path
# Returns: 1 if the folder in which the file is located does not exist
#         0 otherwise
#####
absolutePath() {
    local wdsave absolutePath
    wdsave="$(pwd)"
    ! cd "$(dirname "$1")" 1> /dev/null 2> /dev/null && return 1
    absolutePath="$(pwd)/$(basename "$1")"
    cd "$wdsave"
    echo "$absolutePath"
    return 0
}

# Initialization the configuration parameters with default values
VERBOSITY="$LOG_ERR" # default verbosity level
OKM_URL="http://localhost:8080/OpenKM" # OpenKM URL
OKM_PATH="/okm:root/imported" # Taxonomy path to be imported to

# Parse optional command line arguments
while getopts ":hvgu:p:" opt; do
    case $opt in
        h) usage ; exit 0 ;;
        v) VERBOSITY=$((VERBOSITY+1)) ;;
        g) VERBOSITY="10" ;;
        u) OKM_URL="$OPTARG" ;;
        p) OKM_PATH="$OPTARG" ;;
        \?)
            echo "Invalid option: -$OPTARG" >&2
            usage
            exit $EXIT_BAD_ARGS ;;
        :)
            echo "Option -$OPTARG requires an argument" >&2
            usage
            exit $EXIT_BAD_ARGS ;;
    esac
done

# Remove the optional arguments parsed above.
shift $((OPTIND-1))

# Check if the number of mandatory parameters
# provided is as expected
if [ "$#" -ne "1" ]; then
    echo "Document file name is missing! ($# arguments provided)" >&2
    usage
    exit $EXIT_BAD_ARGS
fi

[ $VERBOSITY -ge $LOG_INFO ] && echo "$TOOLNAME, version: $VERSION"

# Generate the document information
FILENAME=`absolutePath "$1"`
FILENAME_BASE="${FILENAME##*/}"

[ $VERBOSITY -ge $LOG_INFO ] && echo "Document file: '$FILENAME', basename: '$FILENAME_BASE'"

# Logon
[ $VERBOSITY -ge $LOG_DEBUG ] && echo "Logging on to OpenKM Server '$OKM_URL' with '$S"

```

```

response_Auth_logon=$(curl --silent --header "Content-Type: text/xml; charset=UTF-8" \
--header "SOAPAction:action" --data @- \
--request POST "${OKM_URL}/services/OKMAuth" << EOF
  <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:
    <soapenv:Header/>
    <soapenv:Body>
      <ws:login>
        <user>${OKM_ID}</user>
        <password>${OKM_PW}</password>
      </ws:login>
    </soapenv:Body>
  </soapenv:Envelope>
EOF
)

# Check for response code -> Got access token?
token=$(grep -oPm1 "(?<=<return>)[^<]+" <<< "$response_Auth_logon")
if [ -z "$token" ]; then
  echo "Error logging on to OpenKM server!" >&2
  echo "Response: "$response_Auth_logon >&2 && exit $EXIT_BAD_LOGON
else
  [ $VERBOSEITY -ge $LOG_DEBUG ] && echo "Logon successful (Token: '$token')"
fi

# Content must be base64 -> read and encode
[ $VERBOSEITY -ge $LOG_DEBUG ] && echo "Encoding input file as Base64"
content="`base64 "$FILENAME"`"

# Import the document
[ $VERBOSEITY -ge $LOG_DEBUG ] && echo "Importing document '$FILENAME_BASE' to '$OKM_P
response_Doc=$(curl --silent --header "Content-Type: text/xml; charset=UTF-8" \
--header "SOAPAction:action" --data @- \
--request POST "${OKM_URL}/services/OKMDocument" << EOF
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ws=
  <soapenv:Header/>
  <soapenv:Body>
    <ws:createSimple>
      <token>${token}</token>
      <docPath>${OKM_PATH}/${FILENAME_BASE}</docPath>
      <content>${content}</content>
    </ws:createSimple>
  </soapenv:Body>
</soapenv:Envelope>
EOF
)

# Check for success
fault=$(grep -oPm1 "(?<=<faultcode>)[^<]+" <<< "$response_Doc")
if [ -n "$fault" ]; then
  echo "Error importing document: "$fault": ' ' \
    $(grep -oPm1 "(?<=<faultstring>)[^<]+" <<< "$response_Doc") \
    "' -> "$($grep -oPm1 "[^:]+"Exception" <<< "$response_Doc") >&2
  echo "Response: "$response_Doc >&2
else
  uuid=$(grep -oPm1 "(?<=<uuid>)[^<]+" <<< "$response_Doc")
  [ $VERBOSEITY -ge $LOG_DEBUG ] && echo "Document '$uuid' imported successfully"
fi

# Logoff
[ $VERBOSEITY -ge $LOG_DEBUG ] && echo "Logging off from OpenKM server"
response_Auth_logoff=$(curl --silent --header "Content-Type: text/xml; charset=UTF-8" \
--header "SOAPAction:action" --data @- \
--request POST "${OKM_URL}/services/OKMAuth" << EOF
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ws=
  <soapenv:Header/>
  <soapenv:Body>
    <ws:logout>

```

```
<token>${token}</token>
</ws:logout>
</soapenv:Body>
</soapenv:Envelope>
EOF
)

# Check for success
fault=$(grep -oPm1 "(?<=<faultcode>)[^<]+" <<< "$response_Auth_logoff")
if [ -n "$fault" ]; then
    echo "Error logging off from server: \"$fault\": ' " \
        $(grep -oPm1 "(?<=<faultstring>)[^<]+" <<< "$response_Auth_logoff") \
        "' -> "$(grep -oPm1 "[^:]+Exception" <<< "$response_Auth_logoff") >&2
    echo "Response: \"$response_Auth_logoff >&2
else
    [ $VERBOSITY -ge $LOG_DEBUG ] && echo "Logoff successful"
fi

END=`date +%s`
[ $VERBOSITY -ge $LOG_DEBUG ] && echo "Script took $((END-$START)) seconds"

exit $EXIT_OK
```



More information at our public [forum](#).

Creating users from scripts

The script allows you to create users and assign them the user role.

Download file [create-users_okm.xlsx](#)



The string "**newId**" is the **user Id** what will be set.

The string "**newPassword**" is the **user password** what will be set.

The string "**newMail@mail.com**" is the **user mail** what will be set.

The string "**newRealName**" is the **user real name** what will be set.

```
import com.openkm.api.*;

OKMAuth.getInstance().createUser(null, "newId", "newPassword", "newMail@mail.com", "New
OKMAuth.getInstance().assignRole(null, "user1", "ROLE_USER");
```

API description

OpenKM API classes:

OpenKM API class	Description
OKMAuth	Manage security and users. For example, add or remove grants on a node, create or modify users or getting the profiles.
OKMBookmark	Manage the user bookmarks.
OKMDashboard	Manage all data shown on the dashboard.
OKMDocument	Manage documents. For example create, move or delete a document.
OKMFolder	Manage folders. For example create, move or delete a folder.
OKMMail	Manage mails. For example create, move or delete a mails.
OKMNote	Manage notes on any node type. For example create, edit or delete a note on a document, folder, mail or record.
OKMNotification	Manage notifications. For example, add or remove subscriptions on a document or a folder.
OKMProperty	Manage categories and keywords. For example add or remove keywords in a document, folder, mail or record.
OKMPropertyGroup	Manage metadata. For example add metadata group, set metadata fields.
OKMRepository	A lot of stuff related to the repository. For example get the properties of the main root node (/okm:root).
OKMSearch	Manage search feature. For example, manage saved queries or perform a new query to the repository.

<i>OKMStats</i>	<i>General stats of the repository.</i>
<i>OKMUserConfig</i>	<i>Manage user home configuration.</i>
<i>OKMWorkflow</i>	<i>Manage workflows. For example, execute a new workflow.</i>

OKMAuth

Used for managing security and users. For example add or remove grants on a node, create or modify users or getting the profiles.

Basics

The class **com.openkm.bean.Permission** contains permission values (READ, WRITE, etc.). You should use it in combination with methods that are changing or getting security grants.



To set READ and WRITE access you should do:

```
int permission = Permission.READ + Permission.WRITE;
```

To check if you have permission access you should do:

```
// permission is a valid integer value
if ((permission | Permission.WRITE) = Permission.WRITE) {
    // Has WRITE grants.
}
```

On almost methods you'll see parameter named "nodePath". The value of this parameter can be some valid node **path** (folder, document, mail).



Example of nodePath:

- Using path -> "/okm:root/sample.pdf"

Also on all methods you'll see parameter named "**token**". When accessing application across SOAP the login process returns a token, what is used to identify the user on all the exposed methods. From default application execution context you must use "**null**" value what indicates to the application must use the "**user session**".



On special cases you might be "**promoted as Administrator**" using the "**administrator token**".


```
String systemToken = DbSessionManager.getInstance().getSystemToken();
```

Methods


login

Description:

Method	Return values	Description
login()	void	Simulates user UI login process.



When user is logged from UI, are executed some background process what creates main user nodes, like /okm:trash. Unfortunately if the user has never logged from UI and login from API these nodes are still not created and will raise an error, for it is necessary at the beginning execute login method.



The user must be logged before executing the method.

Example:

```
package com.openkm;


import com.openkm.api.OKMAuth;

public class Test {
    public static void main(String[] args) {
        try {
            OKMAuth.getInstance().login();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

login

Description:

Method	Return values	Description
login(String user, String pass)	void	Simulates user UI login process.



When user is logged from UI, are executed some background process what creates main user nodes, like /okm:trash. Unfortunately if the user has never logged from UI and login from API these nodes are still not created and will raise an error, for it is necessary at the beginning execute login method.

Example:

```
package com.openkm;

import com.openkm.api.OKMAuth;

public class Test {
```

```

    public static void main(String[] args) {
        try {
            OKMAuth.getInstance().login("userId", "password");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

logout

Description:

Method	Return values	Description
logout()	void	Kill user session.

Example:

```

package com.openkm;

import com.openkm.api.OKMAuth;

public class Test {
    public static void main(String[] args) {
        try {
            OKMAuth.getInstance().logout();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

grantUser

Description:

Method	Return values	Description
grantUser(String token, String nodePath, String user, int permissions, boolean recursive)	void	Add user grant on a node.
<p>The parameter recursive only has sense when the nodePath is a folder node.</p> <p>When parameter recursive is true, the change will be applied to the node and descendants.</p>		

Example:

```

package com.openkm;

import com.openkm.api.OKMAuth;
import com.openkm.bean.Permission;

public class Test {
    public static void main(String[] args) {
        try {
            // Add john write grants at the node but not descendants
            OKMAuth.getInstance().grantUser(null, "/okm:root", "john", Permission.ALL

            // Add all okmAdmin grants at the node and descendants
            OKMAuth.getInstance().grantUser(null, "/okm:root", "okmAdmin", Permission

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

revokeUser

Description:

Method	Return values	Description
revokeUser(String token, String nodePath, String user, int permissions, boolean recursive)	void	Remove user grant on a node.
<p>The parameter recursive only has sense when the nodePath is a folder node.</p> <p>When parameter recursive is true, the change will be applied to the node and descendants.</p>		

Example:

```

package com.openkm;

import com.openkm.api.OKMAuth;
import com.openkm.bean.Permission;

public class Test {
    public static void main(String[] args) {
        try {
            // Remove john write grants at the node but not descendants
            OKMAuth.getInstance().revokeUser(null, "/okm:root", "john", Permission.AL

            // Remove all okmAdmin grants at the node and descendants
            OKMAuth.getInstance().revokeUser(null, "/okm:root", "okmAdmin", Permissio

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

getGrantedUsers

Description:

Method	Return values	Description
getGrantedUsers(String token, String nodePath)	Map<String, Integer>	Return the granted users of a node.

Example:

```
package com.openkm;

import java.util.Map;
import com.openkm.api.OKMAuth;

public class Test {
    public static void main(String[] args) {
        try {
            Map<String, Integer> grants = OKMAuth.getInstance().getGrantedUsers(null,
                for (String role : grants.keySet()) {
                    System.out.println(role + "->" + grants.get(role));
                }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

grantRole

Description:

Method	Return values	Description
grantRole(String token, String nodePath, String role, int permissions, boolean recursive)	void	Add role grant on a node.
<p>The parameter recursive only has sense when the nodePath is a folder node.</p> <p>When parameter recursive is true, the change will be applied to the node and descendants.</p>		

Example:

```
package com.openkm;

import com.openkm.api.OKMAuth;
import com.openkm.bean.Permission;

public class Test {
    public static void main(String[] args) {
```



```

    try {
        // Add ROLE_USER write grants at the node but not descendants
        OKMAuth.getInstance().grantRole(null, "/okm:root", "ROLE_USER", Permission.WRITE);

        // Add all ROLE_ADMIN grants to the node and descendants
        OKMAuth.getInstance().grantRole(null, "/okm:root", "ROLE_ADMIN", Permission.WRITE);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

revokeRole

Description:

Method	Return values	Description
revokeRole(String token, String nodePath, String role, int permissions, boolean recursive)	void	Remove role grant on a node.
<p>The parameter recursive only has sense when the nodePath is a folder node.</p> <p>When parameter recursive is true, the change will be applied to the node and its descendants.</p>		

Example:

```

package com.openkm;

import com.openkm.api.OKMAuth;
import com.openkm.bean.Permission;

public class Test {
    public static void main(String[] args) {
        try {
            // Remove ROLE_USER write grants at the node but not descendants
            OKMAuth.getInstance().revokeRole(null, "/okm:root", "ROLE_USER", Permission.WRITE);

            // Remove all ROLE_ADMIN grants to the node and descendants
            OKMAuth.getInstance().revokeRole(null, "/okm:root", "ROLE_ADMIN", Permission.WRITE);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

getGrantedRoles

Description:

Method	Return values	Description

<code>getGrantedRoles(String token, String nodePath)</code>	<code>Map<String, Integer></code>	Return the granted roles of a node.
--	--	-------------------------------------

Example:

```
package com.openkm;

import java.util.Map;
import com.openkm.api.OKMAuth;

public class Test {
    public static void main(String[] args) {
        try {
            Map<String, Integer> grants = OKMAuth.getInstance().getGrantedRoles(null,
            for (String role : grants.keySet()) {
                System.out.println(role + "->" + grants.get(role));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

`getUsers`

Description:

Method	Return values	Description
<code>getUsers(String token)</code>	<code>List<String></code>	Return the list of all the users.

Example:

```
package com.openkm;

import com.openkm.api.OKMAuth;

public class Test {
    public static void main(String[] args) {
        try {
            for (String user : OKMAuth.getInstance().getUsers(null)) {
                System.out.println(user);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

`getRoles`

Description:

Method	Return values	Description
--------	---------------	-------------

<i>getRoles(String token)</i>	<i>List<String></i>	<i>Return the list of all the roles.</i>
--------------------------------------	----------------------------------	--

Example:

```
package com.openkm;

import com.openkm.api.OKMAuth;

public class Test {
    public static void main(String[] args) {
        try {
            for (String user : OKMAuth.getInstance().getRoles(null)) {
                System.out.println(user);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getUsersByRole

Description:

<i>Method</i>	<i>Return values</i>	<i>Description</i>
<i>getUsersByRole(String token, String role)</i>	<i>List<String></i>	<i>Return the list of all the users who have assigned a role.</i>

Example:

```
package com.openkm;

import com.openkm.api.OKMAuth;

public class Test {
    public static void main(String[] args) {
        try {
            for (String user : OKMAuth.getInstance().getUsersByRole(null, "ROLE_ADMIN")) {
                System.out.println(user);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getRolesByUser

Description:

<i>Method</i>	<i>Return values</i>	<i>Description</i>

<i>getRolesByUser(String token, String user)</i>	<i>List<String></i>	<i>Return the list of all the roles assigned to a user.</i>
---	----------------------------------	---

Example:

```
package com.openkm;

import com.openkm.api.OKMAuth;

public class Test {
    public static void main(String[] args) {
        try {
            for (String role : OKMAuth.getInstance().getRolesByUser(null, "okmAdmin"))
                System.out.println(role);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getMail

Description:

<i>Method</i>	<i>Return values</i>	<i>Description</i>
<i>getMail(String token, String user)</i>	<i>String</i>	<i>Return the mail of a valid user.</i>

Example:

```
package com.openkm;

import com.openkm.api.OKMAuth;

public class Test {
    public static void main(String[] args) {
        try {
            System.out.println(OKMAuth.getInstance().getMail(null, "okmAdmin"));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getName

Description:

<i>Method</i>	<i>Return values</i>	<i>Description</i>
<i>getName(String token, String user)</i>	<i>String</i>	<i>Return the name of a valid user.</i>

Example:

```
package com.openkm;

import com.openkm.api.OKMAuth;

public class Test {
    public static void main(String[] args) {
        try {
            System.out.println(OKMAuth.getInstance().getName(null, "okmAdmin"));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

changeSecurity

Description:

Method	Return values	Description
public void changeSecurity(String token, String nodePath, Map<String, Integer> grantUsers, Map<String, Integer> revokeUsers, Map<String, Integer> grantRoles, Map<String, Integer> revokeRoles, boolean recursive)	void	Change the security of a node.

Example:

```
package com.openkm;

import java.util.*;
import com.openkm.bean.Permission;
import com.openkm.api.OKMAuth;

public class Test {
    public static void main(String[] args) {
        try {
            String nodePath = "b9736924-bb97-4e2c-8450-138c21e0c9d5";
            Map<String, Integer> grantUsers = new HashMap<>();
            Map<String, Integer> revokeUsers = new HashMap<>();
            Map<String, Integer> grantRoles = new HashMap<>();
            grantRoles.put("ROLE_TEST", Permission.READ | Permission.WRITE);
            Map<String, Integer> revokeRoles = new HashMap<>();
            OKMAuth.getInstance().changeSecurity(null, nodePath, grantUsers, revokeU
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

createUser

Description:

Method	Return values	Description
<i>createUser(String token, String user, String password, String email, String name, boolean active)</i>	void	Create a new user.

Example:

```
package com.openkm;

import com.openkm.api.OKMAuth;

public class Test {
    public static void main(String[] args) {
        try {
            OKMAuth.getInstance().createUser(null, "test", "password.2016", "some@mail.com", "test", true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

deleteUser

Description:

Method	Return values	Description
<i>deleteUser(String token, String user)</i>	void	Delete a user.

Example:

```
package com.openkm;

import com.openkm.api.OKMAuth;

public class Test {
    public static void main(String[] args) {
        try {
            OKMAuth.getInstance().deleteUser(null, "test");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

updateUser

Description:

--

<i>Method</i>	<i>Return values</i>	<i>Description</i>
<i>updateUser(String token, String user, String password, String email, String name, boolean active)</i>	<i>void</i>	Update a user.

Example:

```
package com.openkm;

import com.openkm.api.OKMAuth;

public class Test {
    public static void main(String[] args) {
        try {
            OKMAuth.getInstance().updateUser(null, "test", "newpassword", "some@mail", "test", true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

createRole

Description:

<i>Method</i>	<i>Return values</i>	<i>Description</i>
<i>createRole(String token, String role, boolean active)</i>	<i>void</i>	Create a new role.

Example:

```
package com.openkm;

import com.openkm.api.OKMAuth;

public class Test {
    public static void main(String[] args) {
        try {
            OKMAuth.getInstance().createRole(null, "ROLE_TEST", true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

deleteRole

Description:

<i>Method</i>	<i>Return values</i>	<i>Description</i>

<code>deleteRole(String token, String role)</code>	<code>void</code>	Delete a role.
---	--------------------------	----------------

Example:

```
package com.openkm;

import com.openkm.api.OKMAuth;

public class Test {
    public static void main(String[] args) {
        try {
            OKMAuth.getInstance().deleteRole(null, "ROLE_TEST");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

`updateRole`

Description:

Method	Return values	Description
<code>updateRole(String token, String role, boolean active)</code>	<code>void</code>	Update a role.

Example:

```
package com.openkm;

import com.openkm.api.OKMAuth;

public class Test {
    public static void main(String[] args) {
        try {
            OKMAuth.getInstance().updateRole(null, "ROLE_TEST", true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

`assignRole`

Description:

Method	Return values	Description
<code>assignRole(String token, String user, String role)</code>	<code>void</code>	Assign role to a user.

Example:


```
package com.openkm;  
  
import com.openkm.api.OKMAuth;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            OKMAuth.getInstance().assignRole(null, "test", "ROLE_USER");  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

removeRole

Description:

Method	Return values	Description
removeRole(String token, String user, String role)	void	Remove a role from a user.

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMAuth;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            OKMAuth.getInstance().removeRole(null, "test", "ROLE_USER");  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

OKMBookmark

Used for managing bookmarks of users. For example add or remove bookmark of a node for a user.

Basics

On almost methods you'll see parameter named "**nodePath**". The value of this parameter can be some valid node **path** (folder, document, mail).



Example of nodePath:

- Using path -> "/okm:root/sample.pdf"

Also on all methods you'll see parameter named "**token**". When accessing application across SOAP the login process returns a token, what is used to identify the user on all the exposed methods. From default application execution context you must use "**null**" value what indicates to the application must use the "**user session**".



On special cases you might be "**promoted as Administrator**" using the "**administrator token**".

```
String systemToken = DbSessionManager.getInstance().getSystemToken();
```

Methods

add

Description:

Method	Return values	Description
add(String token, String nodePath, String name)	Bookmark	Add a new bookmark of a node.
The parameter name indicates the name which will be used to store the bookmark. Usually is used the same name of the nodePath.		

Example:

```
package com.openkm;

import com.openkm.api.OKMBookmark;
import com.openkm.dao.bean.Bookmark;

public class Test {
    public static void main(String[] args) {
        try {
```

```

        Bookmark bookmark = OKMBookmark.getInstance().add(null, "/okm:root/documente
        System.out.println(bookmark);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

get

Description:

Method	Return values	Description
get(String token, int bmId)	Bookmark	Gets the bookmark data.
The parameter <i>bmId</i> is the value of the unique bookmark identifier.		

Example:

```

package com.openkm;

import com.openkm.api.OKMBookmark;
import com.openkm.dao.bean.Bookmark;

public class Test {
    public static void main(String[] args) {
        try {
            int bmId = 23;
            Bookmark bookmark = OKMBookmark.getInstance().get(null, bmId);
            System.out.println(bookmark);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

getAll

Description:

Method	Return values	Description
getAll(String token)	List<Bookmark>	Gets all the bookmarks of the logged user.

Example:

```

package com.openkm;

import com.openkm.api.OKMBookmark;
import com.openkm.dao.bean.Bookmark;

```

```

public class Test {
    public static void main(String[] args) {
        try {
            for (Bookmark bookmark : OKMBookmark.getInstance().getAll(null)) {
                System.out.println(bookmark);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

remove

Description:

Method	Return values	Description
remove(String token, int bmId)	void	Delete a bookmark.
The parameter <i>bmId</i> is the value of the unique bookmark identifier.		

Example:

```

package com.openkm;

import com.openkm.api.OKMBookmark;

public class Test {
    public static void main(String[] args) {
        try {
            int bmId = 23;
            OKMBookmark.getInstance().remove(null, bmId);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

rename

Description:

Method	Return values	Description
rename(String token, int bmId, String newName)	Bookmark	Rename a bookmark.
The parameter <i>bmId</i> is the value of the unique bookmark identifier.		

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMBookmark;  
import com.openkm.dao.bean.Bookmark;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            int bmId = 23;  
            Bookmark bookmark = OKMBookmark.getInstance().rename(null, bmId, "newname");  
            System.out.println(bookmark);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

OKMDashboard

Basics

On all methods you'll see parameter named "**token**". When accessing application across SOAP the login process returns a token, what is used to identify the user on all the exposed methods. From default application execution context you must use "**null**" value what indicates to the application must use the "**user session**".



On special cases you might be "**promoted as Administrator**" using the "**administrator token**".

```
String systemToken = DbSessionManager.getInstance().getSystemToken();
```

Methods

getUserCheckedOutDocuments

Description:

Method	Return values	Description
<i>getUserCheckedOutDocuments(String token)</i>	<i>List<DashboardDocumentResult></i>	Get the list of documents checkout by the user.

Example:

```
package com.openkm;

import com.openkm.api.OKMDashboard;
import com.openkm.bean.DashboardDocumentResult;

public class Test {
    public static void main(String[] args) {
        try {
            for (DashboardDocumentResult dashboardDocumentResult : OKMDashboard.getIn
                System.out.println(dashboardDocumentResult);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getUserLastModifiedDocuments

Description:

Method	Return values	Description

<code>getUserLastModifiedDocuments(String token)</code>	<code>List<DashboardDocumentResult></code>	Get the list of the last documents modified by the user.
--	---	--

Example:

```
package com.openkm;

import com.openkm.api.OKMDashboard;
import com.openkm.bean.DashboardDocumentResult;

public class Test {
    public static void main(String[] args) {
        try {
            for (DashboardDocumentResult dashboardDocumentResult : OKMDashboard.getIn
                System.out.println(dashboardDocumentResult);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

`getUserLockedDocuments`

Description:

Method	Return values	Description
<code>getUserLockedDocuments(String token)</code>	<code>List<DashboardDocumentResult></code>	Get the list of the locked documents modified by the user.

Example:

```
package com.openkm;

import com.openkm.api.OKMDashboard;
import com.openkm.bean.DashboardDocumentResult;

public class Test {
    public static void main(String[] args) {
        try {
            for (DashboardDocumentResult dashboardDocumentResult : OKMDashboard.getIn
                System.out.println(dashboardDocumentResult);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

`getUserSubscribedDocuments`

Description:

Method	Return values	Description
<code>getUserSubscribedDocuments(String token)</code>	<code>List<DashboardDocumentResult></code>	Get the list of the subscribed documents by the user.

Example:

```
package com.openkm;

import com.openkm.api.OKMDashboard;
import com.openkm.bean.DashboardDocumentResult;

public class Test {
    public static void main(String[] args) {
        try {
            for (DashboardDocumentResult dashboardDocumentResult : OKMDashboard.getInstance().getSubscribedDocuments(token)) {
                System.out.println(dashboardDocumentResult);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

`getUserSubscribedFolders`

Description:

Method	Return values	Description
<code>getUserSubscribedFolders(String token)</code>	<code>List<DashboardFolderResult></code>	Get the list of the subscribed folders by the user.

Example:

```
package com.openkm;

import com.openkm.api.OKMDashboard;
import com.openkm.bean.DashboardFolderResult;

public class Test {
    public static void main(String[] args) {
        try {
            for (DashboardFolderResult dashboardFolderResult : OKMDashboard.getInstance().getSubscribedFolders(token)) {
                System.out.println(dashboardFolderResult);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```


getUserLastUploadedDocuments

Description:

Method	Return values	Description
<i>getUserLastUploadedDocuments(String token)</i>	<i>List<DashboardDocumentResult></i>	Get the list of the last uploaded documents by the user.

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMDashboard;  
import com.openkm.bean.DashboardDocumentResult;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            for (DashboardDocumentResult dashboardDocumentResult : OKMDashboard.getIn  
                System.out.println(dashboardDocumentResult);  
            }  
            } catch (Exception e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

getUserLastDownloadedDocuments

Description:

Method	Return values	Description
<i>getUserLastDownloadedDocuments(String token)</i>	<i>List<DashboardDocumentResult></i>	Get the list of the last download documents by the user.

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMDashboard;  
import com.openkm.bean.DashboardDocumentResult;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            for (DashboardDocumentResult dashboardDocumentResult : OKMDashboard.getIn  
                System.out.println(dashboardDocumentResult);  
            }  
            } catch (Exception e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

```

        e.printStackTrace();
    }
}

```

getUserLastImportedMails

Description:

Method	Return values	Description
<i>getUserLastImportedMails(String token)</i>	<i>List<DashboardMailResult></i>	Get the list of the last imported mails by the user.

Example:

```

package com.openkm;

import com.openkm.api.OKMDashboard;
import com.openkm.bean.DashboardMailResult;

public class Test {
    public static void main(String[] args) {
        try {
            for (DashboardMailResult dashboardMailResult : OKMDashboard.getInstance().getDashboardMailResults()) {
                System.out.println(dashboardMailResult);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

getUserLastImportedMailAttachments

Description:

Method	Return values	Description
<i>getUserLastImportedMailAttachments(String token)</i>	<i>List<DashboardDocumentResult></i>	Get the list of the last imported mail attachments by the user.

Example:

```

package com.openkm;

import com.openkm.api.OKMDashboard;
import com.openkm.bean.DashboardDocumentResult;

public class Test {
    public static void main(String[] args) {

```

```


        try {
            for (DashboardDocumentResult dashboardDocumentResult : OKMDashboard.getInstance().getDashboardDocumentResults()) {
                System.out.println(dashboardDocumentResult);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

getUserDocumentsSize

Description:

Method	Return values	Description
getUserDocumentsSize(String token)	long	Return the total size in bytes of the user documents.



When user upload or modify a document, the entire size of the contents - and history versions in case of update - are assigned to the users.

Example:

```

package com.openkm;

import com.openkm.api.OKMDashboard;
import com.openkm.util.FormatUtil;


public class Test {
    public static void main(String[] args) {
        try {
            long sizeInBytes = OKMDashboard.getInstance().getUserDocumentsSize(null);
            System.out.println(FormatUtil.formatSize(sizeInBytes));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

getUserSearchs

Description:

Method	Return values	Description
getUserSearchs(String token)	List<QueryParams>	Get the list of all "user news" saved by the user.



The "user saved" searches are executed periodically for getting news based on queries from the repository.

More information at [Working in dashboard](#) and [Working in search](#) views.

Example:

```
package com.openkm;

import com.openkm.api.OKMDashboard;
import com.openkm.dao.bean.QueryParams;

public class Test {
    public static void main(String[] args) {
        try {
            for (QueryParams qp : OKMDashboard.getInstance().getUserSearchs(null)) {
                System.out.println(qp);
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

find

Description:

Method	Return values	Description
<i>find(String token, int qpId)</i>	<i>List<DashboardNodeResult></i>	<i>Execute a "user news".</i>
The param qpId is the unique identifier of the "users news" search.		

Example:

```
package com.openkm;

import com.openkm.api.OKMDashboard;
import com.openkm.bean.DashboardNodeResult;
import com.openkm.bean.Document;
import com.openkm.bean.Folder;
import com.openkm.bean.Mail;

public class Test {
    public static void main(String[] args) {
        try {
            int qpId = 12;
            for (DashboardNodeResult dashboardNodeResult : OKMDashboard.getInstance().
                if (dashboardNodeResult.getNode() instanceof Document ) {
                    Document doc = (Document) dashboardNodeResult.getNode();
                    System.out.println(doc);
                } else if (dashboardNodeResult.getNode() instanceof Folder ) {
                    Folder fld = (Folder) dashboardNodeResult.getNode();
                    System.out.println(fld);
                } else if (dashboardNodeResult.getNode() instanceof Mail ) {
```

```

        Mail mail = (Mail) dashboardNodeResult.getNode();
        System.out.println(mail);
    }
}

} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

getLastWeekTopDownloadedDocuments

Description:

Method	Return values	Description
<i>getLastWeekTopDownloadedDocuments(String token)</i>	<i>List<DashboardDocumentResult></i>	Get the list of the top downloaded documents during the last week by all the users.

Example:

```

package com.openkm;

import com.openkm.api.OKMDashboard;
import com.openkm.bean.DashboardDocumentResult;

public class Test {
    public static void main(String[] args) {
        try {
            for (DashboardDocumentResult dashboardDocumentResult : OKMDashboard.getIn
                System.out.println(dashboardDocumentResult);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

getLastMonthTopDownloadedDocuments

Description:

Method	Return values	Description
<i>getLastMonthTopDownloadedDocuments(String token)</i>	<i>List<DashboardDocumentResult></i>	Get the list of the top downloaded documents during the last month by all the users.

Example:

```
package com.openkm;

import com.openkm.api.OKMDashboard;
import com.openkm.bean.DashboardDocumentResult;

public class Test {
    public static void main(String[] args) {
        try {
            for (DashboardDocumentResult dashboardDocumentResult : OKMDashboard.getIn
                System.out.println(dashboardDocumentResult);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getLastWeekTopModifiedDocuments

Description:

Method	Return values	Description
<i>getLastWeekTopModifiedDocuments(String token)</i>	<i>List<DashboardDocumentResult></i>	Get the list of the top modified documents during the last week by all the users.

Example:

```
package com.openkm;

import com.openkm.api.OKMDashboard;
import com.openkm.bean.DashboardDocumentResult;

public class Test {
    public static void main(String[] args) {
        try {
            for (DashboardDocumentResult dashboardDocumentResult : OKMDashboard.getIn
                System.out.println(dashboardDocumentResult);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getLastMonthTopModifiedDocuments

Description:

Method	Return values	Description

<code>getLastMonthTopModifiedDocuments(String token)</code>	<code>List<DashboardDocumentResult></code>	Get the list of the top modified documents during the last month by all the users.
--	---	--

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMDashboard;  
import com.openkm.bean.DashboardDocumentResult;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            for (DashboardDocumentResult dashboardDocumentResult : OKMDashboard.getIn  
                System.out.println(dashboardDocumentResult);  
            }  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

`getLastModifiedDocuments`

Description:

Method	Return values	Description
<code>getLastModifiedDocuments(String token)</code>	<code>List<DashboardDocumentResult></code>	Get the list of the last modified documents by all the users.

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMDashboard;  
import com.openkm.bean.DashboardDocumentResult;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            for (DashboardDocumentResult dashboardDocumentResult : OKMDashboard.getIn  
                System.out.println(dashboardDocumentResult);  
            }  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

`getLastUploadedDocuments`

Description:

Method	Return values	Description
<code>getLastUploadedDocuments(String token)</code>	<code>List<DashboardDocumentResult></code>	Get the list of the last uploaded documents by all the users.

Example:

```
package com.openkm;

import com.openkm.api.OKMDashboard;
import com.openkm.bean.DashboardDocumentResult;

public class Test {
    public static void main(String[] args) {
        try {
            for (DashboardDocumentResult dashboardDocumentResult : OKMDashboard.getIn
                System.out.println(dashboardDocumentResult);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

visiteNode

Description:

Method	Return values	Description
<code>visiteNode(String token, String source, String node, Calendar date)</code>	<code>void</code>	Mark a node as a visited.



From UI, the nodes are shown strong until are visited by the user. This method mark the node as visited by the user in the database.

Available sources:

- `LastWeekTopDownloadedDocuments`
- `LastMonthTopDownloadedDocuments`
- `LastWeekTopModifiedDocuments`
- `LastMonthTopModifiedDocuments`
- `LastModifiedDocuments`
- `LastUploadedDocuments`

- *UserLockedDocuments*
- *UserCheckedOutDocuments*
- *UserLastModifiedDocuments*
- *UserLastDownloadedDocuments*
- *UserSubscribedDocuments*
- *UserSubscribedFolders*
- *UserLastUploadedDocuments*
- *UserLastImportedMails*
- *UserLastImportedMailAttachments*
- *In case "user new", the source name is the name of the query.*

Example:

```
package com.openkm;  
  
import java.util.Calendar;  
  
import com.openkm.api.OKMDashboard;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            Calendar cal = Calendar.getInstance();  
            OKMDashboard.getInstance().visiteNode(null, "LastWeekTopDownloadedDocuments");  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

OKMDocument

Basics

On almost methods you'll see parameter named "**docId**". The value of this parameter can be some valid document **UUID** or **path** node.



Example of docId:

- Using UUID -> "c41f9ea0-0d6c-45da-bae4-d72b66f42d0f";
- Using path -> "/okm:root/sample.pdf"

When you'll see parameter named "**fldId**". The value of this parameter can be some valid folder **UUID** or **path** node.



Example of docId:

- Using UUID -> "c6785440-0d6c-45da-1234-d9874563d0f";
- Using path -> "/okm:root/folder1"

About the parameter named "**dstId**", the value of this parameter can be some valid folder **UUID** or **path** node.



Example of dstId:

- Using UUID -> "c4455667-0d6c-45da-1234-d98564323e0";
- Using path -> "/okm:root/folder2"



When is used parameter named "**docPath**" then the value of this parameter can be some valid document .

Example of docPath:

- Using path -> "/okm:root/sample.pdf"

Also on all methods you'll see parameter named "**token**". When accessing application across SOAP the login process returns a token, what is used to identify the user on all the exposed methods. From default application execution context you must use "**null**" value what indicates to the application must use the "**user session**".



On special cases you might be "**promoted as Administrator**" using the "**administrator token**".

```
String systemToken = DbSessionManager.getInstance().getSystemToken();
```


Methods

create

Description:

Method	Return values	Description
<code>create(String token, Document doc, InputStream is)</code>	Document	Create a new document.

The parameter `doc` must be initialized with the OpenKM path value into the repository. Optionally you can also set keywords and title values.

 The other parameteres are not taken in consideration for document creation.

Example:

```
package com.openkm;

import java.io.FileInputStream;
import java.io.InputStream;

import org.apache.commons.io.IOUtils;

import com.openkm.api.OKMDocument;
import com.openkm.bean.Document;

public class Test {
    public static void main(String[] args) {
        InputStream is = null;
        try {
            is = new FileInputStream("/home/user/sample.pdf");
            Document doc = new Document();
            doc.setPath("/okm:root/sample.pdf");
            // Optional
            doc.setTitle("sample document");
            doc.getKeywords().add("key1");
            doc.getKeywords().add("key2");
            // Create document
            doc = OKMDocument.getInstance().create(null, doc, is);
            System.out.println(doc);
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            IOUtils.closeQuietly(is);
        }
    }
}
```

createSimple

Description:

Method	Return values	Description
<i>createSimple(String token, String docPath, InputStream is)</i>	Document	Create a new document.
The parameter docPath must be initialized with the OpenKM path value into the repository.		

Example:

```
package com.openkm;

import java.io.FileInputStream;
import java.io.InputStream;

import org.apache.commons.io.IOUtils;

import com.openkm.api.OKMDocument;
import com.openkm.bean.Document;

public class Test {
    public static void main(String[] args) {
        InputStream is = null;
        try {
            is = new FileInputStream("/home/user/sample.pdf");
            String docPath = "/okm:root/sample.pdf";
            Document doc = OKMDocument.getInstance().createSimple(null, docPath, is);
            System.out.println(doc);
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            IOUtils.closeQuietly(is);
        }
    }
}
```

delete

Description:

Method	Return values	Description
<i>delete(String token, String docId)</i>	void	Delete a document.

Example:

```
package com.openkm;

import com.openkm.api.OKMDocument;

public class Test {
    public static void main(String[] args) {
        try {
            // Delete document by path
            OKMDocument.getInstance().delete(null, "/okm:root/sample.pdf");
            // Delete document by uuid
            OKMDocument.getInstance().delete(null, "adc048d7-3791-4540-ae32-680629f73
```

```
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

getProperties

Description:

Method	Return values	Description
getProperties(String token, String docId)	Document	Return a document properties.

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMDocument;  
import com.openkm.bean.Document;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            Document doc = OKMDocument.getInstance().getProperties(null, "adc048d7-37  
                System.out.println(doc);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

getContent

Description:

Method	Return values	Description
getContent(String token, String docId, boolean checkout)	InputStream	Return the document content.

When paramer checkout is true, the document is marked for edition.

Example:

```
package com.openkm;  
  
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.InputStream;  
import java.io.OutputStream;
```

```

import org.apache.commons.io.IOUtils;

import com.openkm.api.OKMDocument;

public class Test {
    public static void main(String[] args) {
        InputStream is = null;
        OutputStream fos = null;
        try {
            is = OKMDocument.getInstance().getContent(null, "adc048d7-3791-4540-ae32-4
            File tmp = new File("/home/user/sample.pdf");
            fos = new FileOutputStream(tmp);
            IOUtils.copy(is, fos);
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            IOUtils.closeQuietly(is);
            IOUtils.closeQuietly(fos);
        }
    }
}

```

getContentByVersion

Description:

Method	Return values	Description
getContentByVersion(String token, String docId, String versionId)	InputStream	Return the document content by version.
The parameter versionIs is the name of the document version.		

Example:

```

package com.openkm;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;

import org.apache.commons.io.IOUtils;

import com.openkm.api.OKMDocument;

public class Test {
    public static void main(String[] args) {
        InputStream is = null;
        OutputStream fos = null;
        try {
            is = OKMDocument.getInstance().getContentByVersion(null, "adc048d7-3791-4
            File tmp = new File("/home/user/sample.pdf");
            fos = new FileOutputStream(tmp);
            IOUtils.copy(is, fos);
        } catch (Exception e) {

```


```

        e.printStackTrace();
    } finally {
        IOUtils.closeQuietly(is);
        IOUtils.closeQuietly(fos);
    }
}

```

getChilds

Description:

Method	Return values	Description
getChilds(String token, String fldId)	List<Document>	Returns a list of all documents which their parent is fldId.
The parameter fldId can be a folder node.		
 This method is deprecated in favour of getChildren method. We encourage do not using it, because on nearly future will be removed.		

Example:

```

package com.openkm;

import com.openkm.api.OKMDocument;
import com.openkm.bean.Document;

public class Test {
    public static void main(String[] args) {
        try {
            for (Document doc : OKMDocument.getInstance().getChilds(null, "/okm:root/")
                System.out.println(doc);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

getChildren

Descripcion:

Method	Return values	Description
getChildren(String token, String fldId)	List<Document>	Returns a list of all documents which their parent is fldId.

The parameter *fldId* can be a folder node.

Example:

```
package com.openkm;

import com.openkm.api.OKMDocument;
import com.openkm.bean.Document;

public class Test {
    public static void main(String[] args) {
        try {
            for (Document doc : OKMDocument.getInstance().getChildren(null, "/okm:root")) {
                System.out.println(doc);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

rename

Description:

Method	Return values	Description
rename(String token, String docId, String newName)	Document	Changes the name of a document.

Example:

```
package com.openkm;


import com.openkm.api.OKMDocument;
import com.openkm.bean.Document;

public class Test {
    public static void main(String[] args) {
        try {
            Document doc = OKMDocument.getInstance().rename(null, "/okm:root/samples.1");
            System.out.println(doc);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

setProperties

Description:

Method	Return values	Description

<i>setProperties(String token, Document doc)</i>	<i>void</i>	<i>Changes some document properties.</i>
<p>Variables allowed to be changed:</p> <ul style="list-style-type: none"> • Title • Description • Language • Associated categories • Associated keywords <div style="border: 1px dashed orange; padding: 10px; margin-top: 10px;">  Only not null and not empty variables will be taken in consideration. </div>		

Example:

```
package com.openkm;

import com.openkm.api.OKMDocument;
import com.openkm.bean.Document;

public class Test {
    public static void main(String[] args) {
        try {
            Document doc = OKMDocument.getInstance().getProperties(null, "/okm:root/s.
            doc.getKeywords().add("key1");
            doc.setDescription("some description");
            OKMDocument.getInstance().setProperties(null, doc);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

checkout

Description:

Method	Return values	Description
<i>checkout(String token, String docId)</i>	<i>void</i>	<i>Marks the document for edition.</i>
<p>Only one user can modify a document at a time.</p> <p>Before starting edition you must do a checkout action that locks the edition process for other users and allows edition only to the user who has executed the action.</p>		

Example:


```
package com.openkm;

import com.openkm.api.OKMDocument;

public class Test {
    public static void main(String[] args) {
        try {
            OKMDocument.getInstance().checkout(null, "/okm:root/samples.pdf");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

cancelCheckout

Description:

Method	Return values	Description
cancelCheckout(String token, String docId)	void	Cancels a document edition.
<div>  This action can only be done by the user who previously executed the checkout action. </div>		

Example:

```
package com.openkm;

import com.openkm.api.OKMDocument;

public class Test {
    public static void main(String[] args) {
        try {
            OKMDocument.getInstance().cancelCheckout(null, "/okm:root/samples.pdf");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

forceCancelCheckout

Description:

Method	Return values	Description
forceCancelCheckout(String token, String docId)	void	Cancels a document edition.

This method allows to cancel edition on any document.

It is not mandatory to execute this action by the same user who previously executed the checkout action.



This action can only be done by a super user (user with ROLE_ADMIN).

Example:

```
package com.openkm;

import com.openkm.api.OKMDocument;

public class Test {
    public static void main(String[] args) {
        try {
            OKMDocument.getInstance().forceCancelCheckout(null, "/okm:root/samples.pdf");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

isCheckedOut

Description:

Method	Return values	Description
isCheckedOut(String token, String docId)	void	Returns a boolean that indicates if the document is on edition or not.

Example:


```
package com.openkm;

import com.openkm.api.OKMDocument;

public class Test {
    public static void main(String[] args) {
        try {
            System.out.println(OKMDocument.getInstance().isCheckedOut(null, "/okm:root/samples.pdf"));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

checkin

Description:

Method	Return values	Description
<i>checkin(String token, String docId, InputStream is, String comment)</i>	Version	Updates a document with a new version and returns an object with new Version values.
<div>  Only the user who started the edition - checkout - is allowed to update the document. </div>		

Example:

```
package com.openkm;

import java.io.FileInputStream;
import java.io.InputStream;


import org.apache.poi.util.IOUtils;

import com.openkm.api.OKMDocument;

public class Test {
    public static void main(String[] args) {
        InputStream is = null;
        try {
            is = new FileInputStream("/home/user/sample.pdf");
            OKMDocument.getInstance().checkin(null, "/okm:root/samples.pdf", is, "opt
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            IOUtils.closeQuietly(is);
        }
    }
}
```

checkin

Description:

Method	Return values	Description
<i>checkin(String token, String docId, InputStream is, String comment, int increment)</i>	Version	Updates a document with a new version and returns an object with new Version values.
The increment value depends on VersionNumeration implementation.		
<div>  More information about VersionNumeration at Creating your own Version Number Adapter. </div>		



Only the user who started the edition - checkout - is allowed to update the document.

Example:

```
package com.openkm;

import java.io.FileInputStream;
import java.io.InputStream;

import org.apache.poi.util.IOUtils;

import com.openkm.api.OKMDocument;
import com.openkm.vernum.VersionNumerationAdapter;

public class Test {
    public static void main(String[] args) {
        InputStream is = null;
        try {
            is = new FileInputStream("/home/user/sample.pdf");
            OKMDocument.getInstance().checkin(null, "/okm:root/samples.pdf", is, "opt
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            IOUtils.closeQuietly(is);
        }
    }
}
```

isCheckedOut

Description:

Method	Return values	Description
isCheckedOut(String token, String docId)	void	Returns a boolean that indicates if the document is on edition or not.

Example:

```
package com.openkm;

import com.openkm.api.OKMDocument;

public class Test {
    public static void main(String[] args) {
        try {
            System.out.println(OKMDocument.getInstance().isCheckedOut(null, "/okm:roo
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
}

```

getVersionHistory

Description:

Method	Return values	Description
getVersionHistory(String token, String docId	List<Version>	Returns a list of all document versions.

Example:

```
package com.openkm;

import com.openkm.api.OKMDocument;
import com.openkm.bean.Version;

public class Test {
    public static void main(String[] args) {
        try {
            for (Version ver :OKMDocument.getInstance().getVersionHistory(null, "/okm
                System.out.println(ver);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

lock

Description:

Method	Return values	Description
lock(String token, String docId	LockInfo	Locks a document and returns an object with the Lock information.



Only the user who locked the document is allowed to unlock.

A locked document cannot be modified by other users.

Example:

```
package com.openkm;

import com.openkm.api.OKMDocument;
import com.openkm.bean.LockInfo;

public class Test {
    public static void main(String[] args) {
```


```

        try {
            LockInfo lockInfo = OKMDocument.getInstance().lock(null, "/okm:root/sample.pdf");
            System.out.println(lockInfo);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

unlock

Description:

Method	Return values	Description
unlock(String token, String docId	LockInfo	Unlocks a locked document.
 Only the user who locked the document is allowed to unlock.		

Example:

```

package com.openkm;

import com.openkm.api.OKMDocument;

public class Test {
    public static void main(String[] args) {
        try {
            OKMDocument.getInstance().unlock(null, "/okm:root/samples.pdf");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

forceUnlock

Description:

Method	Return values	Description
forceUnlock(String token, String docId	LockInfo	Unlocks a locked document.
<p>This method allows to unlock any locked document.</p> <p>It is not mandatory execute this action by the same user who previously executed the checkout lock action.</p> <p>This action can only be done by a super user (user with <code>ROLE_ADMIN</code>).</p>		



Example:

```
package com.openkm;

import com.openkm.api.OKMDocument;

public class Test {
    public static void main(String[] args) {
        try {
            OKMDocument.getInstance().forceUnlock(null, "/okm:root/samples.pdf");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

isLocked

Description:

Method	Return values	Description
isLocked(String token, String docId)	void	Returns a boolean that indicates if the document is locked or not.

Example:

```
package com.openkm;

import com.openkm.api.OKMDocument;

public class Test {
    public static void main(String[] args) {
        try {
            System.out.println(OKMDocument.getInstance().isLocked(null, "/okm:root/sa
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getLockInfo

Description:

Method	Return values	Description
getLockInfo(String token, String docId)	LockInfo	Returns an object with the Lock information.

Example:

```
package com.openkm;

import com.openkm.api.OKMDocument;
import com.openkm.bean.LockInfo;


public class Test {
    public static void main(String[] args) {
        try {
            LockInfo lockInfo = OKMDocument.getInstance().getLockInfo(null, "/okm:root");
            System.out.println(lockInfo);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

purge

Description:

Method	Return values	Description
<code>purge(String token, String docId)</code>	void	The document is definitely removed from repository.

Usually you will purge documents into `/okm:trash/userId` - the personal trash user locations - but is possible to directly purge any document from the whole repository.



When a document is purged only will be able to be restored from a previously repository backup. The purge action removes the document definitely from the repository.

Example:

```
package com.openkm;

import com.openkm.api.OKMDocument;

public class Test {
    public static void main(String[] args) {
        try {
            OKMDocument.getInstance().purge(null, "/okm:root/samples.pdf");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

move

Description:

Method	Return values	Description
<code>move(String token, String docId, String dstId)</code>	<code>void</code>	Move document into some folder.
The values of the <code>dstId</code> parameter should be a folder UUID or path.		

Example:


```
package com.openkm;

import com.openkm.api.OKMDocument;

public class Test {
    public static void main(String[] args) {
        try {
            OKMDocument.getInstance().move(null, "/okm:root/samples.pdf", "/okm/root/");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

copy

Description:

Method	Return values	Description
<code>copy(String token, String docId, String dstId)</code>	<code>void</code>	Copy a document into some folder.
The values of the <code>dstId</code> parameter should be a folder UUID or path.		
<div>  <p>Only the binary data and the security grants are copied to destination, the metadata, keywords, etc. of the document are not copied.</p> <p>See "extendedCopy" method for this feature.</p> </div>		

Example:

```
package com.openkm;

import com.openkm.api.OKMDocument;

public class Test {
    public static void main(String[] args) {
        try {
            OKMDocument.getInstance().copy(null, "/okm:root/samples.pdf", "/okm/root/");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
}
}
```

extendedCopy

Description:

Method	Return values	Description
<i>extendedCopy(String token, String docId, String dstId, String docName, ExtendedAttributes extAttr)</i>	void	Copies a document with associated data into some folder.

The values of the *dstId* parameter should be a folder UUID or path.

When the parameter *newName* value is null, the document will preserve the same name.

i By default only the binary data and the security grants, the metadata, keywords, etc. of the document are not copied.

Additional:

- When the *category* parameter is true the original values of the categories will be copied.
- When the *keywords* parameter is true the original values of the keywords will be copied.
- When the *propertyGroups* parameter is true the original values of the metadata groups will be copied.
- When the *notes* parameter is true the original value of the notes will be copied.
- When *wiki* parameter is true the original value of the wiki will be copied.

Example:

```
package com.openkm;

import com.openkm.api.OKMDocument;
import com.openkm.bean.ExtendedAttributes;

public class Test {
    public static void main(String[] args) {
        try {
            ExtendedAttributes extAttr = new ExtendedAttributes();
            // copy keywords
            extAttr.setKeywords(true);
            OKMDocument.getInstance().extendedCopy(null, "/okm:root/samples.pdf", "/o.
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

restoreVersion

Description:

Method	Return values	Description
restoreVersion(String token, String docId, String versionId)	void	Restores previously document version to actual version.

Example:

```
package com.openkm;

import com.openkm.api.OKMDocument;

public class Test {
    public static void main(String[] args) {
        try {
            OKMDocument.getInstance().restoreVersion(null, "/okm:root/samples.pdf", "
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

purgeVersionHistory

Description:

Method	Return values	Description
purgeVersionHistory(String token, String docId)	void	Purges all the previous document versions except the actual version.

Example:

```
package com.openkm;

import com.openkm.api.OKMDocument;

public class Test {
    public static void main(String[] args) {
        try {
            OKMDocument.getInstance().purgeVersionHistory(null, "/okm:root/samples.pdf
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getVersionHistorySize

Description:

Method	Return values	Description
getVersionHistorySize(String token, String docId)	long	Returns the sum in bytes of all documents into documents history.

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMDocument;  
import com.openkm.util.FormatUtil;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            System.out.println(FormatUtil.formatSize(OKMDocument.getInstance().getVersionHistorySize(token, docId)));  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

isValid

Description:

Method	Return values	Description
isValid(String token, String docId)	boolean	Returns a boolean that indicates if the node is a document or not.

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMDocument;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            boolean valid = OKMDocument.getInstance().isValid(token, docId);  
            System.out.println(valid);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
    }  
}
```

getPath

Description:

<i>Method</i>	<i>Return values</i>	<i>Description</i>
<i>getPath(String token, String uuid)</i>	<i>String</i>	<i>Converts a document UUID to document path.</i>

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMDocument;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            String docPath = OKMDocument.getInstance().getPath(null, "f123a950-0329-4  
            System.out.println(docPath);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

OKMFolder

Basics

On almost methods you'll see parameter named "**fldId**". The value of this parameter can be some valid folder **UUID** or **path**.



Example of fldId:

- Using UUID -> "c52f9ea0-0d6c-45da-bae4-d72b66f42da3";
- Using path -> "/okm:root/test"

About the parameter named "**dstId**", the value of this parameter can be some valid folder **UUID** or **path** node.



Example of dstId:

- Using UUID -> "c4455667-0d6c-45da-1234-d98564323e0";
- Using path -> "/okm:root/folder2"

Also on all methods you'll see parameter named "**token**". When accessing application across SOAP the login process returns a token, what is used to identify the user on all the exposed methods. From default application execution context you must use "**null**" value what indicates to the application must use the "**user session**".



On special cases you might be "**promoted as Administrator**" using the "**administrator token**".

```
String systemToken = DbSessionManager.getInstance().getSystemToken();
```

Methods

create

Description:

Method	Return values	Description
create(String token, Folder fld)	Folder	Creates a new folder and returns as a result an object Folder.
<p>The variable path into the parameter fld, must be initialized. It indicates the folder path into OpenKM.</p> <pre>Folder fld = new Folder(); fld.setPath("/okm:root/test");</pre>		



The other variables of the Folder (fld) will not take any effect on the folder creation.

We suggest using the method below to create the Folder Simply rather this one.

Example:

```
package com.openkm;

import com.openkm.api.OKMFolder;
import com.openkm.bean.Folder;

public class Test {
    public static void main(String[] args) {
        try {
            Folder fld = new Folder();
            fld.setPath("/okm:root/test");
            OKMFolder.getInstance().create(null, fld);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

createSimple

Description:

Method	Return values	Description
createSimple(String token, String fldPath)	Folder	Creates a new folder and returns as a result an object Folder.

Example:

```
package com.openkm;

import com.openkm.api.OKMFolder;

public class Test {
    public static void main(String[] args) {
        try {
            String fldPath = "/okm:root/test";
            OKMFolder.getInstance().createSimple(null, fldPath);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getProperties

Description:

Method	Return values	Description
<i>getProperties(String token, String fldId)</i>	Folder	Return the folder properties.

Example:

```
package com.openkm;

import com.openkm.api.OKMFolder;
import com.openkm.bean.Folder;

public class Test {
    public static void main(String[] args) {
        try {
            Folder fld = OKMFolder.getInstance().getProperties(null, "/okm:root/test");
            System.out.println(fld);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

delete

Description:

Method	Return values	Description
<i>delete(String token, String fldId)</i>	void	Delete a folder.

Example:

```
package com.openkm;

import com.openkm.api.OKMFolder;

public class Test {
    public static void main(String[] args) {
        try {
            OKMFolder.getInstance().delete(null, "/okm:root/test");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

purge

Description:

Method	Return values	Description

purge(String token, String fldId)***void****The folder is definitely removed from the repository.*

Usually you will purge folders into `/okm:trash/userId` - the personal trash user locations - but it is possible to directly purge any folder from the whole repository.



When a folder is purged, it will only be able to be restored from a previously repository backup. The purge action removes the folder definitely from the repository.

Example:

```
package com.openkm;

import com.openkm.api.OKMFolder;

public class Test {
    public static void main(String[] args) {
        try {
            OKMFolder.getInstance().purge(null, "/okm:root/test");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

rename

Description:

Method	Return values	Description
<i>rename(String token, String fldId, String newName)</i>	<i>Folder</i>	<i>Rename a folder.</i>

Example:

```
package com.openkm;

import com.openkm.api.OKMFolder;

public class Test {
    public static void main(String[] args) {
        try {
            OKMFolder.getInstance().rename(null, "/okm:root/test", "newname");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

move

Description:

Method	Return values	Description
<code>move(String token, String fldId, String dstId)</code>	void	Moves folder into some folder.
The values of the <code>dstId</code> parameter should be a folder UUID or path.		

Example:


```
package com.openkm;

import com.openkm.api.OKMFolder;

public class Test {
    public static void main(String[] args) {
        try {
            OKMFolder.getInstance().move(null, "/okm:root/test", "/okm:root/move/dest");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

copy

Description:

Method	Return values	Description
<code>copy(String token, String fldId, String dstId)</code>	void	Copies a folder into a folder.
The values of the <code>dstId</code> parameter should be a folder UUID or path.		
 Only the security grants are copied to destination, the metadata, keywords, etc. of the folder are not copied. See " <code>extendedCopy</code> " method for this feature.		

Example:

```
package com.openkm;

import com.openkm.api.OKMFolder;

public class Test {
    public static void main(String[] args) {
        try {
            OKMFolder.getInstance().copy(null, "/okm:root/test", "/okm:root/move/dest");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```

        e.printStackTrace();
    }
}

```

extendedCopy

Description:

Method	Return values	Description
<i>extendedCopy(String token, String fldId, String dstId, ExtendedAttributes extAttr)</i>	void	Copies a folder with the associated data into some folder.

The values of the *dstId* parameter should be a folder UUID or path.

i By default only the binary data and the security grants, the metadata, keywords, etc. of the folder are not copied.

Additional extended attributes parameters:

- When the *category* parameter is true the original values of the categories will be copied.
- When the *keywords* parameter is true the original values of the keywords will be copied.
- When the *propertyGroups* parameter is true the original values of the metadata groups will be copied.
- When the *notes* parameter is true the original values of the notes will be copied.
- When the *wiki* parameter is true the original values of the wiki will be copied.

Example:

```

package com.openkm;

import com.openkm.api.OKMFolder;
import com.openkm.bean.ExtendedAttributes;

public class Test {
    public static void main(String[] args) {
        try {
            ExtendedAttributes extAttr = new ExtendedAttributes();
            extAttr.setKeywords(true);
            OKMFolder.getInstance().extendedCopy(null, "/okm:root/sample.pdf", "/okm:root/sample.pdf");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

extendedCopy

Description:

Method	Return values	Description
<i>extendedCopy(String token, String fldId, String dstId, ExtendedAttributes extAttr, String newName)</i>	void	Copies a folder with the associated data into some folder.
<p>The values of the <i>dstId</i> parameter should be a folder UUID or path.</p> <p>When parameter <i>newName</i> value is null, folder will preservate the same name.</p> <div> <p>i By default only the binary data and the security grants, the metadata, keywords, etc. of the folder are not copied.</p> <p>Additional extended attributes parameters:</p> <ul style="list-style-type: none"> • When the <i>category</i> parameter is true the original values of the categories will be copied. • When the <i>keywords</i> parameter is true the original values of the keywords will be copied. • When the <i>propertyGroups</i> parameter is true the original values of the metadata groups will be copied. • When the <i>notes</i> parameter is true the original values of the notes will be copied. • When the <i>wiki</i> parameter is true the original values of the wiki will be copied. </div>		

Example:

```
package com.openkm;


import com.openkm.api.OKMFolder;
import com.openkm.bean.ExtendedAttributes;

public class Test {
    public static void main(String[] args) {
        try {
            ExtendedAttributes extAttr = new ExtendedAttributes();
            extAttr.setKeywords(true);
            OKMFolder.getInstance().extendedCopy(null, "/okm:root/sample.pdf", "/okm:r

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getChilds

Description:

Method	Return values	Description
<i>getChilds(String token, String fldId)</i>	<i>List<Folder></i>	Returns a list of all folders which their parent is fldId.
The parameter fldId can be a folder node.		
 This method is deprecated in favour of getChildren method. We encourage do not using it, because on nearly future will be removed.		

Example:

```
package com.openkm;

import com.openkm.api.OKMFolder;
import com.openkm.bean.Folder;

public class Test {
    public static void main(String[] args) {
        try {
            for (Folder fld : OKMFolder.getInstance().getChilds(null, "/okm:root/test")) {
                System.out.println(fld);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getChildren

Description:

Method	Return values	Description
<i>getChildren(String token, String fldId)</i>	<i>List<Folder></i>	Returns a list of all folders which their parent is fldId.
The parameter fldId can be a folder node.		

Example:

```
package com.openkm;

import com.openkm.api.OKMFolder;
import com.openkm.bean.Folder;

public class Test {
    public static void main(String[] args) {
```

```

        try {
            for (Folder fld : OKMFolder.getInstance().getChildren(null, "/okm:root/te
                System.out.println(fld);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

getContentInfo

Description:

Method	Return values	Description
<i>getContentInfo(String token, String fldId)</i>	<i>ContentInfo</i>	Return and object ContentInfo with information about folder.
<p>The ContentInfo object retrieves information about:</p> <ul style="list-style-type: none"> • The number of folders into. • The number of documents into. • The number of mails into. • The size in bytes of all objects into the folder. 		

Example:

```

package com.openkm;

import com.openkm.api.OKMFolder;
import com.openkm.bean.ContentInfo;

public class Test {
    public static void main(String[] args) {
        try {
            ContentInfo ci = OKMFolder.getInstance().getContentInfo(null, "/okm:root
            System.out.println(ci);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

isValid

Description:

Method	Return values	Description
--------	---------------	-------------

<i>isValid(String token, String fldId)</i>	<i>boolean</i>	<i>Returns a boolean that indicates if the node is a folder or not.</i>
---	-----------------------	---

Example:

```
package com.openkm;

import com.openkm.api.OKMFolder;

public class Test {
    public static void main(String[] args) {
        try {
            boolean valid = OKMFolder.getInstance().isValid(null, "/okm:root/test");
            System.out.println(valid);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getPath

Description:

<i>Method</i>	<i>Return values</i>	<i>Description</i>
<i>getPath(String token, String uuid)</i>	<i>String</i>	<i>Convert folder UUID to folder path.</i>

Example:

```
package com.openkm;

import com.openkm.api.OKMFolder;

public class Test {
    public static void main(String[] args) {
        try {
            String path = OKMFolder.getInstance().getPath(null, "c52f9ea0-0d6c-45da-");
            System.out.println(path);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

createMissingFolders

Description:

<i>Method</i>	<i>Return values</i>	<i>Description</i>
<i>createMissingFolders(String token, String</i>		<i>Create missing folders based on all the path</i>

fldPath)***void****nodes.*

The method checks all subfolders and when detects some one is missing, it creates all missing subfolders from there.

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMFolder;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            OKMFolder.getInstance().createMissingFolders(null, "/okm:root/test/missin  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

OKMMail

Basics

On almost methods you'll see parameter named "**mailId**". The value of this parameter can be some valid mail **UUID** or **path**.



Example of mailId:

- Using UUID -> "c52f9ea0-0d6c-45da-bae4-d72b66f42da3";
- Using path -> "/okm:root/2937b81d-0b10-4dd0-a426-9acbd80be1c9-some subject"

About the parameter named "**dstId**", the value of this parameter can be some valid folder **UUID** or **path** node.



Example of dstId:

- Using UUID -> "c4455667-0d6c-45da-1234-d98564323e0";
- Using path -> "/okm:root/folder2"

About the parameter named "**fldId**", the value of this parameter can be some valid folder **UUID** or **path** node.



Example of fldId:

- Using UUID -> "c52f9ea0-0d6c-45da-bae4-d72b66f42da3";
- Using path -> "/okm:root/test"

Also on all methods you'll see parameter named "**token**". When accessing application across SOAP the login process returns a token, what is used to identify the user on all the exposed methods. From default application execution context you must use "**null**" value what indicates to the application must use the "**user session**".



On special cases you might be "**promoted as Administrator**" using the "**administrator token**".

```
String systemToken = DbSessionManager.getInstance().getSystemToken();
```

Methods

create

Description:

Method	Return values	Description

<code>create(String token, Mail mail)</code>	Mail	Creates a new mail and returns as a result an object Mail.
<p>The variable path into the parameter mail, must be initialized. It indicates the folder path into OpenKM.</p> <p>Other mandatory variables:</p> <ul style="list-style-type: none"> • size (mail size in bytes). • from (mail from account). • reply, to, cc, bcc (mail accounts are optional). • sendDate (date when mail was sent). • receivedDate (date when was received). • subject (mail subject). • content (the mail content). • mimeType (HTML or text mime type). <div style="border: 1px dashed #add8e6; padding: 10px; margin: 10px 0;"> <p>i Mai accounts allowed formats:</p> <ul style="list-style-type: none"> • "\"John King\" <jking@mail.com>" • "<jking@mail.com>" </div> <div style="border: 1px dashed #add8e6; padding: 10px; margin: 10px 0;"> <p>i Mail path allowed is:</p> <p>MSGID + "-" + sanitized(subject).</p> </div> <div style="border: 1px dashed #add8e6; padding: 10px; margin: 10px 0;"> <p>i MIME types values:</p> <ul style="list-style-type: none"> • Mail.MIME_TEXT for text mail format. • Mail.MIME_HTML for html mail format. </div>		

Example:

```
package com.openkm;

import java.util.Arrays;
import java.util.Calendar;

import org.owasp.encoder.Encode;

import com.openkm.api.OKMMail;
import com.openkm.bean.Mail;

public class Test {
    public static void main(String[] args) {
        try {
```

```

        Mail mail = new Mail();

        // Mail path = msgId + escaped(subject)
        String msgId = "2937b81d-0b10-4dd0-a426-9acbd80be1c9";
        String subject = "some subject";
        String mailPath = "/okm:root/" + msgId + "-" + escape(subject);
        mail.setPath(mailPath);

        // Other format for mail "some name <no_reply@openkm.com>"
        mail.setFrom("<no_reply@openkm.com>");
        mail.setTo((String[])Arrays.asList("anonymous@gmail.com").toArray());

        // You should set real dates
        mail.setSentDate(Calendar.getInstance());
        mail.setReceivedDate(Calendar.getInstance());
        mail.setContent("some content");
        mail.setMimeType(Mail.MIME_TEXT);
        mail.setSubject(subject);

        // Get only as an approximation of real size for these sample
        mail.setSize(mail.toString().getBytes("UTF-8").length);
        OKMMail.getInstance().create(null, mail);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

private static String escape(String name) {
    String ret = cleanup(name);

    // Fix XSS issues
    ret = Encode.forHtml(ret);

    return ret;
}

private static String cleanup(String name) {
    String ret = name.replace("/", "");
    ret = ret.replace("*", "");
    ret = ret.replaceAll("\\s+", " ").trim();
    return ret;
}
}

```

getProperties

Description:

Method	Return values	Description
getProperties(String token, String mailId)	Mail	Returns the mail properties.

Example:

```

package com.openkm;

import com.openkm.api.OKMMail;
import com.openkm.bean.Mail;

public class Test {

```

```

    public static void main(String[] args) {
        try {
            Mail mail = OKMMail.getInstance().getProperties(null, "064ff51a-b815-4f48
            System.out.println(mail);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

delete

Description:

Method	Return values	Description
<i>delete(String token, String mailId)</i>	void	Delete a mail.

Example:

```

package com.openkm;

import com.openkm.api.OKMMail;
import com.openkm.bean.Folder;

public class Test {
    public static void main(String[] args) {
        try {
            OKMMail.getInstance().delete(null, "064ff51a-b815-4f48-a096-b4946876784f"
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

purge

Description:

Method	Return values	Description
<i>purge(String token, String mailId)</i>	void	The mail is definitely removed from the repository.

Usually you will purge mails into /okm:trash/userId - the personal trash user locations - but it is possible to directly purge any mail from the whole repository.



When a mail is purged, it will only be able to be restored from a previously repository backup. The purge action removes the mail definitely from the repository.

Example:

```
package com.openkm;

import com.openkm.api.OKMMail;

public class Test {
    public static void main(String[] args) {
        try {
            OKMMail.getInstance().purge(null, "064ff51a-b815-4f48-a096-b4946876784f")
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

rename

Description:

Method	Return values	Description
rename(String token, String mailId, String newName)	Mail	Rename a mail.

Example:

```
package com.openkm;

import com.openkm.api.OKMMail;
import com.openkm.bean.Folder;

public class Test {
    public static void main(String[] args) {
        try {
            OKMMail.getInstance().rename(null, "064ff51a-b815-4f48-a096-b4946876784f", "NewName")
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

move

Description:

Method	Return values	Description
move(String token, String mailId, String dstId)	void	Moves mail into some folder.
The values of the dstId parameter should be a folder UUID or path.		

Example:


```
package com.openkm;

import com.openkm.api.OKMMail;
import com.openkm.bean.Folder;

public class Test {
    public static void main(String[] args) {
        try {
            OKMMail.getInstance().move(null, "064ff51a-b815-4f48-a096-b4946876784f",
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

copy

Description:

Method	Return values	Description
copy(String token, String mailId, String dstId)	void	Copies a mail into a folder.
The values of the dstId parameter should be a folder UUID or path.		
<div>  Only the security grants are copied to destination, the metadata, keywords, etc. of the mail are not copied. See "extendedCopy" method for this feature. </div>		

Example:

```
package com.openkm;


import com.openkm.api.OKMMail;
import com.openkm.bean.Folder;

public class Test {
    public static void main(String[] args) {
        try {
            OKMMail.getInstance().copy(null, "064ff51a-b815-4f48-a096-b4946876784f",
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

extendedCopy

Description:

--

Method	Return values	Description
<i>extendedCopy(String token, String mailId, String dstId, ExtendedAttributes extAttr)</i>	void	Copies a mail with the associated data into some folder.
<p>The values of the <i>dstId</i> parameter should be a folder UUID or path.</p> <div>  <p>By default only the binary data and the security grants, the metadata, keywords, etc. of the mail are not copied.</p> <p>Additional extended attributes parameters:</p> <ul style="list-style-type: none"> • When the <i>category</i> parameter is true the original values of the categories will be copied. • When the <i>keywords</i> parameter is true the original values of the keywords will be copied. • When the <i>propertyGroups</i> parameter is true the original values of the metadata groups will be copied. • When the <i>notes</i> parameter is true the original values of the notes will be copied. • When the <i>wiki</i> parameter is true the original values of the wiki will be copied. </div>		

Example:

```
package com.openkm;

import com.openkm.api.OKMMail;
import com.openkm.bean.ExtendedAttributes;

public class Test {
    public static void main(String[] args) {
        try {
            ExtendedAttributes extAttr = new ExtendedAttributes();
            extAttr.setKeywords(true);
            OKMMail.getInstance().extendedCopy(null, "064ff51a-b815-4f48-a096-b494687

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getChilds

Description:

Method	Return values	Description
<i>getChilds(String token, String fldId)</i>	List<Mail>	Returns a list of all mails which their parent is mailId.

The parameter `mailId` can be a folder node.



This method is deprecated in favour of `getChildren` method. We encourage do not using it, because on nearly future will be removed.

Example:

```
package com.openkm;

import com.openkm.api.OKMMail;
import com.openkm.bean.Mail;

public class Test {
    public static void main(String[] args) {
        try {
            for (Mail mail : OKMMail.getInstance().getChilids(null, "/okm:root/test"))
                System.out.println(mail);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

`getChildren`

Description:

Method	Return values	Description
<code>getChildren(String token, String fldId)</code>	<code>List<Mail></code>	Returns a list of all mails which their parent is <code>mailId</code> .

The parameter `mailId` can be a folder node.

Example:

```
package com.openkm;

import com.openkm.api.OKMMail;
import com.openkm.bean.Mail;

public class Test {
    public static void main(String[] args) {
        try {
            for (Mail mail : OKMMail.getInstance().getChildren(null, "/okm:root/test"))
                System.out.println(mail);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
}
```

isValid

Description:

Method	Return values	Description
isValid(String token, String mailId)	boolean	Returns a boolean that indicates if the node is a mail or not.

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMMail;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            boolean valid = OKMMail.getInstance().isValid(null, "064ff51a-b815-4f48-a0  
            System.out.println(valid);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

getPath

Description:

Method	Return values	Description
getPath(String token, String uuid)	String	Convert folder UUID to mail path.

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMMail;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            String path = OKMMail.getInstance().getPath(null, "064ff51a-b815-4f48-a0  
            System.out.println(path);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

importEml

Description:

Method	Return values	Description
importEml(String path, InputStream is)	Mail	Import a mail in EML format.
The values of the path parameter should be a folder path. The path parameter indicate where the mail will be stored in the repository after is imported.		

Example:

```
package com.openkm;  
  
import java.io.FileInputStream;  
import java.io.InputStream;  
  
import org.apache.commons.io.IOUtils;  
  
import com.openkm.api.OKMMail;  
import com.openkm.bean.Mail;  
  
public class Test {  
    public static void main(String[] args) {  
        InputStream is = null;  
        try {  
            is = new FileInputStream("/home/files/test.eml");  
            Mail mail = OKMMail.getInstance().importEml("d88cff0d-903a-4c5a-82ea-8e6d");  
            System.out.println(mail);  
        } catch (Exception e) {  
            e.printStackTrace();  
        } finally {  
            IOUtils.closeQuietly(is);  
        }  
    }  
}
```

importMsg

Description:

Method	Return values	Description
importMsg(String path, InputStream is)	Mail	Import a mail in MSG format.
The values of the path parameter should be a folder path. The path parameter indicate where the mail will be stored in the repository after is imported.		

Example:

```
package com.openkm;  
  
import java.io.FileInputStream;  
import java.io.InputStream;  
  
import org.apache.commons.io.IOUtils;  
  
import com.openkm.api.OKMMail;  
import com.openkm.bean.Mail;  
  
public class Test {  
    public static void main(String[] args) {  
        InputStream is = null;  
        try {  
            is = new FileInputStream("/home/files/test.msg");  
            Mail mail = OKMMail.getInstance().importMsg("d88cff0d-903a-4c5a-82ea-8e6d");  
            System.out.println(mail);  
        } catch (Exception e) {  
            e.printStackTrace();  
        } finally {  
            IOUtils.closeQuietly(is);  
        }  
    }  
}
```

OKMNote

Basics

On most methods you'll see parameter named "**nodeId**". The value of this parameter can be a valid document, folder, or mail **UUID** or **path**.



Example of nodeId:

- Using UUID -> "f123a950-0329-4d62-8328-0ff500fd42db";
- Using path -> "/okm:root/logo.png"

Also on all methods you'll see parameter named "**token**". When accessing application across SOAP the login process returns a token, what is used to identify the user on all the exposed methods. From default application execution context you must use "**null**" value what indicates to the application must use the "**user session**".



On special cases you might be "**promoted as Administrator**" using the "**administrator token**".

```
String systemToken = DbSessionManager.getInstance().getSystemToken();
```

Methods

addNote

Description:

Method	Return values	Description
add(String token, String nodeId, String text)	Note	Adds a note to a node and returns an object Note.

Example:

```
package com.openkm;


import com.openkm.api.OKMNote;
import com.openkm.bean.Note;

public class Test {
    public static void main(String[] args) {
        try {
            Note note = OKMNote.getInstance().add(null, "/okm:root/logo.png", "the no
            System.out.println(note);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
}
```

get

Description:


Method	Return values	Description
<i>get(String token, String noteId)</i>	<i>Note</i>	<i>Retrieves the note.</i>
<div> <i>The noteId is an UUID.</i> <i>The object Node has a variable named path, in that case the path contains an UUID.</i></div>		

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMNote;  
import com.openkm.bean.Note;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            for (Note note : OKMNote.getInstance().list(null, "/okm:root/logo.png"))  
                Note data = OKMNote.getInstance().get(null, note.getPath());  
                System.out.println(data);  
            }  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

delete

Description:

Method	Return values	Description
<i>delete(String token, String noteId)</i>	<i>void</i>	<i>Delete a note.</i>
<div> <i>The noteId is an UUID.</i> <i>The object Node has a variable named path, in that case the path contains an UUID.</i></div>		

Example:

```

package com.openkm;

import com.openkm.api.OKMNote;
import com.openkm.bean.Note;


public class Test {
    public static void main(String[] args) {
        try {
            for (Note note : OKMNote.getInstance().list(null, "/okm:root/logo.png"))
                OKMNote.getInstance().delete(null, note.getPath());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

set

Description:

Method	Return values	Description
set(String token, String noteId, String text)	void	Changes the note text.



The noteId is an UUID.

The object Node has a variable named path, in that case the path contains an UUID.

Example:

```

package com.openkm;

import com.openkm.api.OKMNote;
import com.openkm.bean.Note;

public class Test {
    public static void main(String[] args) {
        try {
            for (Note note : OKMNote.getInstance().list(null, "/okm:root/logo.png"))
                OKMNote.getInstance().set(null, note.getPath());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

list

Description:

Method	Return values	Description
--------	---------------	-------------

<i>list(String token, String nodeId)</i>	<i>List<Note></i>	<i>Retrieves a list of all the notes of a node.</i>
---	--------------------------------	---

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMNote;  
import com.openkm.bean.Note;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            for (Note note : OKMNote.getInstance().list(null, "/okm:root/logo.png"))  
                OKMNote.getInstance().set(null, note.getPath());  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```


OKMNotification

Basics

On most methods you'll see parameter named "**nodeId**". The value of this parameter can be a valid document, folder, or mail **UUID** or **path** node.



Example of nodeId:

- Using path -> `"/okm:root/logo.png"`
- Using uuid -> `"33584fe7-01fc-4ba8-883b-be6979796093"`

About the parameter named "**nodePath**", the value of this parameter can be a valid document, folder, or mail **path** node.



Example of nodePath:

- Using path -> `"/okm:root/logo.png"`

Also on all methods you'll see parameter named "**token**". When accessing application across SOAP the login process returns a token, what is used to identify the user on all the exposed methods. From default application execution context you must use "**null**" value what indicates to the application must use the "**user session**".



On special cases you might be "**promoted as Administrator**" using the "**administrator token**".

```
String systemToken = DbSessionManager.getInstance().getSystemToken();
```

Methods

subscribe

Description:

Method	Return values	Description
<code>subscribe(String token, String nodePath)</code>	<code>void</code>	Adds a subscription to a node.

Example:

```
package com.openkm;

import com.openkm.api.OKMNotification;
```

```

public class Test {
    public static void main(String[] args) {
        try {
            OKMNotification.getInstance().subscribe(null, "/okm:root/logo.png");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

unsubscribe

Description:

Method	Return values	Description
unsubscribe(String token, String nodePath)	void	Delete a subscription to a node.

Example:

```

package com.openkm;

import com.openkm.api.OKMNotification;

public class Test {
    public static void main(String[] args) {
        try {
            OKMNotification.getInstance().unsubscribe(null, "/okm:root/logo.png");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

getSubscribers

Description:

Method	Return values	Description
getSubscribers(String token, String nodePath)	Set<String>	Retrieves a list of all the subscribers of a node.

Example:

```

package com.openkm;

import com.openkm.api.OKMNotification;

public class Test {
    public static void main(String[] args) {
        try {
            for (String subscriber : OKMNotification.getInstance().getSubscribers(token, nodePath)) {
                System.out.println(subscriber);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

    }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

notify

Description:

Method	Return values	Description
<i>notify(String token, String nodeId, List<String> users, List<String> mails, String message, boolean attachment)</i>	void	Send a mail notification.

i The parameter users are a set of OpenKM users to be notified.

The parameter mails are a set of mails - usually external mails - to be notified.

The parameter message is the content body of the mail.

When attachment value is true the node is attached into the mail.

Example:

```

package com.openkm;

import java.util.ArrayList;
import java.util.List;

import com.openkm.api.OKMNotification;

public class Test {
    public static void main(String[] args) {
        try {
            List<String> users = new ArrayList<>();
            users.add("user1");
            List<String> mails = new ArrayList<>();
            String body = "Body of the message";
            OKMNotification.getInstance().notify(null, "/okm:root/logo.png", users, m
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}


```

proposedSubscription

Description:

Return

Method	values	Description
<i>proposedSubscription(String token, String nodeId, List<String> users, String comment)</i>	void	Propose a subscription by a mail notification.



The parameter users are a set of OpenKM users to be notified.

The parameter comment is the content body of the mail.

Example:

```
package com.openkm;

import java.util.ArrayList;
import java.util.List;

import com.openkm.api.OKMNotification;

public class Test {
    public static void main(String[] args) {
        try {
            List<String> users = new ArrayList<>();
            users.add("user1");
            String comment = "Body of the message";
            OKMNotification.getInstance().proposedSubscription(null, "/okm:root/logo.",
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

OKMProperty

Basics

On most methods you'll see parameter named "**nodeId**". The value of this parameter can be a valid document, folder, mail **UUID** or **path** node.



Example of nodeId:

- Using UUID -> "f123a950-0329-4d62-8328-0ff500fd42db";
- Using path -> "/okm:root/logo.png"

About the parameter named "**catId**", the value of this parameter can be a valid categories folder **UUID** or **path** node.



Example of nodeId:

- Using UUID -> "f1678435-0329-4d62-8328-09846e6739823";
- Using path -> "/okm:categories/category1"

About the parameter named "**nodePath**", the value of this parameter can be a valid document, folder, or mail **path** node.



Example of nodePath:

- Using path -> "/okm:root/logo.png"

Also on all methods you'll see parameter named "**token**". When accessing application across SOAP the login process returns a token, what is used to identify the user on all the exposed methods. From default application execution context you must use "**null**" value what indicates to the application must use the "**user session**".



On special cases you might be "**promoted as Administrator**" using the "**administrator token**".

```
String systemToken = DbSessionManager.getInstance().getSystemToken();
```

Methods

addCategory

Description:

Method	Return values	Description

<code>addCategory(String token, String nodeId, String catId)</code>	<code>void</code>	Set a relation between a category and a node.
The value of the <code>catId</code> parameter should be a category folder UUID or path.		

Example:

```
package com.openkm;

import com.openkm.api.OKMProperty;

public class Test {
    public static void main(String[] args) {
        try {
            OKMProperty.getInstance().addCategory(null, "/okm:root/logo.png", "/okm:c
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

`removeCategory`

Description:

Method	Return values	Description
<code>removeCategory(String token, String nodeId, String catId)</code>	<code>void</code>	Removes a relation between a category and a node.
The value of the <code>catId</code> parameter should be a category folder UUID or path.		

Example:


```
package com.openkm;

import com.openkm.api.OKMProperty;

public class Test {
    public static void main(String[] args) {
        try {
            OKMProperty.getInstance().removeCategory(null, "/okm:root/logo.png", "/ok
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

`addKeyword`

Description:

Method	Return values	Description
<code>addKeyword(String token, String nodeId, String keyword)</code>	String	Add a keyword in a node.
<p>The keyword should be a single word without spaces, formats allowed:</p> <ul style="list-style-type: none"> • "test" • "two_words" (the character "_" is used for the junction). <div>  We also we suggest you to add keyword in lowercase format, because OpenKM is case sensitive. </div>		

Example:

```
package com.openkm;

import com.openkm.api.OKMProperty;

public class Test {
    public static void main(String[] args) {
        try {
            String keyword = OKMProperty.getInstance().addKeyword(null, "/okm:root/logo.png", "test");
            System.out.println(keyword);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

removeKeyword

Description:

Method	Return values	Description
<code>removeKeyword(String token, String nodeId, String keyword)</code>	void	Removes a keyword from a node.

Example:

```
package com.openkm;

import com.openkm.api.OKMProperty;

public class Test {
    public static void main(String[] args) {
        try {
            OKMProperty.getInstance().removeKeyword(null, "/okm:root/logo.png", "test");
        }
    }
}
```


```

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

setEncryption

Description:

Method	Return values	Description
setEncryption(String token, String nodePath, String cipherName)	void	Marks a document as an encrypted binary data in the repository.
<p>The parameter <i>nodeId</i> should be a document node.</p> <p>The parameter <i>cipherName</i> saves information about the encryption mechanism.</p> <div>  <p>This method does not perform any kind of encryption, simply mark in the database that a document is encrypted.</p> </div>		

Example:

```

package com.openkm;

import com.openkm.api.OKMProperty;

public class Test {
    public static void main(String[] args) {
        try {
            OKMProperty.getInstance().setEncryption(null, "/okm:root/logo.png", "phra
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

unsetEncryption

Description:

Method	Return values	Description
unsetEncryption(String token, String nodePath)	void	Marks a document is a normal binary data into repository.

The parameter `nodeId` should be a document node.



This method does not perform any kind of encryption, simply mark into the database that a document has been unencrypted.

Example:

```
package com.openkm;

import com.openkm.api.OKMProperty;

public class Test {
    public static void main(String[] args) {
        try {
            OKMProperty.getInstance().unsetEncryption(null, "/okm:root/logo.png");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

setSigned

Description:

Method	Return values	Description
setSigned(String token, String nodePath, boolean signed)	void	Marks a document as signed or unsigned binary data into the repository.

The parameter `nodeId` should be a document node.



This method does not perform any kind of digital signature process, simply mark into the database that a document is signed.

Example:

```
package com.openkm;

import com.openkm.api.OKMProperty;

public class Test {
    public static void main(String[] args) {
        try {
            OKMProperty.getInstance().setSigned(null, "/okm:root/logo.png", true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
}  
}
```

OKMPropertyGroup

Basics



From the older OpenKM version we named "**Metadata Groups**" as "**Property Groups**".

Although we understand this name does not help a lot to identify these methods with metadata ones, for historical reason, we continue maintaining the nomenclature.

For more information about [Metadata](#).

On most methods you'll see a parameter named "**nodeId**". The value of this parameter can be a valid document, folder, mail or record **UUID** or **path**.



Example of nodeId:

- Using UUID -> "**f123a950-0329-4d62-8328-0ff500fd42db**";
- Using path -> "**/okm:root/logo.png**"



The class `com.openkm.util.ISO8601` should be used to set and parse metadata date fields. The metadata field of type date values is stored in the application in ISO-8601 basic format.

To convert the retrieved metadata field of type date to a valid date use:

```
Calendar cal = ISO8601.parseBasic(metadataFieldValue);
```

To save the date value in the metadata field of type date use:

```
Calendar cal = Calendar.getInstance(); // Present date
String metadataFieldValue = ISO8601.formatBasic(cal);
// metadataFieldValue can be saved into repository metadata field of type date
```

Also on all methods you'll see parameter named "**token**". When accessing application across SOAP the login process returns a token, what is used to identify the user on all the exposed methods. From default application execution context you must use "**null**" value what indicates to the application must use the "**user session**".



On special cases you might be "**promoted as Administrator**" using the "**administrator token**".

```
String systemToken = DbSessionManager.getInstance().getSystemToken();
```

Methods

addGroup

Description:

Method	Return values	Description
<code>addGroup(String token, String nodeId, String grpName)</code>	<code>void</code>	Adds an empty metadata group to a node.
The <code>grpName</code> should be a valid Metadata group name.		

Example:

```
package com.openkm;

import com.openkm.api.OKMPropertyGroup;

public class Test {
    public static void main(String[] args) {
        try {
            OKMPropertyGroup.getInstance().addGroup(null, "/okm:root/logo.png", "okg:");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

`removeGroup`

Description:

Method	Return values	Description
<code>removeGroup(String token, String nodeId, String grpName)</code>	<code>void</code>	Removes a metadata group of a node.
The <code>grpName</code> should be a valid Metadata group name.		

Example:

```
package com.openkm;

import com.openkm.api.OKMPropertyGroup;

public class Test {
    public static void main(String[] args) {
        try {
            OKMPropertyGroup.getInstance().removeGroup(null, "/okm:root/logo.png", "o.
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getGroups

Description:

Method	Return values	Description
getGroups(String token, String nodeId)	List<PropertyGroup>	Retrieves a list of metadata groups assigned to a node.

Example:

```
package com.openkm;

import com.openkm.api.OKMPropertyGroup;
import com.openkm.bean.PropertyGroup;

public class Test {
    public static void main(String[] args) {
        try {
            for (PropertyGroup pg : OKMPropertyGroup.getInstance().getGroups(null, "/"))
                System.out.println(pg);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getAllGroups

Description:

Method	Return values	Description
getAllGroups(String token)	List<PropertyGroup>	Retrieves a list of all metadata groups set into the application.

Example:


```
package com.openkm;

import com.openkm.api.OKMPropertyGroup;
import com.openkm.bean.PropertyGroup;

public class Test {
    public static void main(String[] args) {
        try {
            for (PropertyGroup pg : OKMPropertyGroup.getInstance().getAllGroups(null))
                System.out.println(pg);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getProperties

Description:

Method	Return values	Description
getProperties(String token, String nodeId, String grpName)	List<FormElement>	Retrieves a list of all metadata group elements and its node values.
The grpName should be a valid Metadata group name.		
 The method is usually used to display form elements with its values to be shown or changed by used.		

Example:


```
package com.openkm;

import com.openkm.api.OKMPropertyGroup;
import com.openkm.bean.form.FormElement;

public class Test {
    public static void main(String[] args) {
        try {
            for (FormElement fe : OKMPropertyGroup.getInstance().getProperties(null, "
                System.out.println(fe);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getPropertiesSimple

Description:

Method	Return values	Description
getPropertiesSimple(String token, String nodeId, String grpName)	Map<String, String>	Retrieves a list of all metadata group elements and its node values.
The grpName should be a valid Metadata group name.		
 The method is usually used to display form elements with its values to be shown or changed by used.		

Example:

```
package com.openkm;

import java.util.Map;

import com.openkm.api.OKMPropertyGroup;

public class Test {
    public static void main(String[] args) {
        try {
            Map<String, String> properties = OKMPropertyGroup.getInstance().getProperties();
            for (String key : properties.keySet()) {
                System.out.println(key + ":" + properties.get(key));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

setProperties

Description:

Method	Return values	Description
setProperties(String token, String nodeId, String grpName, List<FormElement> properties)	void	Changes the metadata node.

The grpName should be a valid Metadata group name.

i It is not mandatory to set in the parameter ofeList all FormElement, it is enough with the formElements you wish to set values.

i The sample below is based on this Metadata group definition:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.0//EN"
    "http://www.openkm.com/dtd/property-groups"
<property-groups>
  <property-group label="Consulting" name="okg:consulting">
    <input label="Name" type="text" name="okp:consulting.name"/>
    <input label="Date" type="date" name="okp:consulting.date" />
    <checkbox label="Important" name="okp:consulting.important"/>
    <textarea label="Comment" name="okp:consulting.comment"/>
  </property-group>
</property-groups>
```

Example:

```
package com.openkm;

import java.util.ArrayList;
import java.util.List;

import com.openkm.api.OKMPropertyGroup;
import com.openkm.bean.form.FormElement;
import com.openkm.bean.form.Input;

public class Test {
    public static void main(String[] args) {
        try {
            List<FormElement> fElements = OKMPropertyGroup.getInstance().getProperties();
            for (FormElement fElement : fElements) {
                if (fElement.getName().equals("okp:consulting.name")) {
                    Input name = (Input) fElement;
                    name.setValue("new value");
                }
            }

            OKMPropertyGroup.getInstance().setProperties(null, "/okm:root/logo.pdf",

            // Same modification with only affected FormElement
            fElements = new ArrayList<>();
            Input name = new Input();
            name.setName("okp:consulting.name");
            name.setValue("new value");
            fElements.add(name);
            OKMPropertyGroup.getInstance().setProperties(null, "/okm:root/logo.pdf",

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

setPropertySimple

Description:

Method	Return values	Descri
setPropertySimple(String token, String nodeId, String propGroup, String propName, String propValue)	void	Changes the metadata of a node.
The grpName should be a valid Metadata group name.		
<div><div>i</div><div>It is not mandatory to set in the properties parameter all fields values, is enough with the fields you wish to change</div></div>		
<div><div>i</div><div>The sample below is based on this Metadata group definition:</div></div>		


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.0//EN"
    "http://www.openkm.com/dtd/property-groups"

<property-groups>
  <property-group label="Consulting" name="okg:consulting">
    <input label="Name" type="text" name="okp:consulting.name"/>
    <input label="Date" type="date" name="okp:consulting.date" />
    <checkbox label="Important" name="okp:consulting.important"/>
    <textarea label="Comment" name="okp:consulting.comment"/>
  </property-group>
</property-groups>
```



To add several values on a metadata field of type multiple like this:

```
<select label="Multiple" name="okp:consulting.multiple" type="multiple">
  <option label="One" value="one"/>
  <option label="Two" value="two"/>
  <option label="Three" value="three" />
</select>
```

You should do:

```
"one;two";
```

Where **"one"** and **"two"** are valid values and character **";"** is used as separator.

Example:

```
package com.openkm;

import com.openkm.api.OKMPropertyGroup;

public class Test {
    public static void main(String[] args) {
        try {
            OKMPropertyGroup.getInstance().setPropertySimple(null, "/okm:root/logo.pd
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

setPropertiesSimple

Description:

Method	Return values	Descr
setPropertiesSimple(String token, String nodeId, String grpName, Map<String, String> properties)	void	Changes the metadata of a node.

The `grpName` should be a valid Metadata group name.



It is not mandatory to set in the `properties` parameter all fields values, is enough with the fields you wish to change



The sample below is based on this Metadata group definition:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.0//EN"
    "http://www.openkm.com/dtd/property-groups"
<property-groups>
  <property-group label="Consulting" name="okg:consulting">
    <input label="Name" type="text" name="okp:consulting.name"/>
    <input label="Date" type="date" name="okp:consulting.date" />
    <checkbox label="Important" name="okp:consulting.important"/>
    <textarea label="Comment" name="okp:consulting.comment"/>
  </property-group>
</property-groups>
```



To add several values on a metadata field of type multiple like this:

```
<select label="Multiple" name="okp:consulting.multiple" type="multiple">
  <option label="One" value="one"/>
  <option label="Two" value="two"/>
  <option label="Three" value="three" />
</select>
```

You should do:

```
properties.put("okp:consulting.multiple", "one;two");
```

Where **"one"** and **"two"** are valid values and character **";"** is used as separator.

Example:

```
package com.openkm;

import java.util.Calendar;
import java.util.HashMap;
import java.util.Map;

import com.openkm.api.OKMPropertyGroup;
import com.openkm.util.ISO8601;

public class Test {
    public static void main(String[] args) {
        try {
            Map<String, String> properties = new HashMap<>();
            properties.put("okp:consulting.name", "new name");
```


```

        // Date fields must be saved with basic ISO 8601 format
        properties.put("okp:consulting.date", ISO8601.formatBasic(Calendar.getInstance().getTime()));
        OKMPropertyGroup.getInstance().setPropertiesSimple(null, "/okm:root/logo.");
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

getPropertyGroupForm

Description:

Method	Return values	Description
getPropertyGroupForm(String token, String grpName)	List<FormElement>	Retrieves a list of all metadata group elements definition.
The grpName should be a valid Metadata group name.		
 The method is usually used to display empty form elements for creating new metadata values.		

Example:

```

package com.openkm;

import com.openkm.api.OKMPropertyGroup;
import com.openkm.bean.form.FormElement;

public class Test {
    public static void main(String[] args) {
        try {
            for (FormElement fe : OKMPropertyGroup.getInstance().getPropertyGroupForm("token", "grpName")) {
                System.out.println(fe);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

hasGroup

Description:

Method	Return values	Description
hasGroup(String token, String nodeId, String grpName)	boolean	Returns a boolean that indicate if the node has or not a metadata group.

Example:

```
package com.openkm;

import com.openkm.api.OKMPropertyGroup;

public class Test {
    public static void main(String[] args) {
        try {
            boolean found = OKMPropertyGroup.getInstance().hasGroup(null, "/okm:root/");
            System.out.println(found);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getSuggestions

Description:

Method	Return values	Description
<code>getSuggestions(String token, String nodeId, String grpName, String propName)</code>	<code>List<String></code>	Retrieves a list of a suggested values.

The propName parameter should be a [Metadata Select field](#) type.



More information at [cCreating your own Suggestion Analyzer](#) and [Metadata Select field](#).



The sample below is based on this Metadata group definition:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.0//EN"
    "http://www.openkm.com/dtd/property-groups">
<property-groups>
  <property-group label="Technology" name="okg:technology">
    <select label="Type" name="okp:technology.type" type="multiple">
      <option label="Alfa" value="t1"/>
      <option label="Beta" value="t2" />
      <option label="Omega" value="t3" />
    </select>
    <select label="Language" name="okp:technology.language" type="simple">
      <option label="Java" value="java"/>
      <option label="Python" value="python"/>
      <option label="PHP" value="php" />
    </select>
    <input label="Comment" name="okp:technology.comment"/>
    <textarea label="Description" name="okp:technology.description"/>
  </property-group>
</property-groups>
```

```
<input label="Link" type="link" name="okp:technology.link"/>
</property-group>
</property-groups>
```

Example:


```
package com.openkm;

import com.openkm.api.OKMPropertyGroup;

public class Test {
    public static void main(String[] args) {
        try {
            for (String value : OKMPropertyGroup.getInstance().getSuggestions(null, "")) {
                System.out.println(value);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

registerDefinition

Description:

Method	Return values	Description
registerDefinition(String token, String pgDef)	void	Set the XML Metada groups definition into the \$TOMCAT_HOME/PropertyGroups.xml file.
<div>  <p>The method only can be executed by super user granted (ROLE_ADMIN member user).</p> <p>Take in mind chaging metadata is done in two steps, first modifying PropertyGroups.xml file and then registering the changing. For the second step take a loot at registerPropertyGroups method in OKMRepository class for it.</p> </div>		

Example:

```
package com.openkm;

import java.io.FileInputStream;
import java.io.InputStream;

import org.apache.commons.io.IOUtils;

import com.openkm.api.OKMPropertyGroup;

public class Test {
```

```
public static void main(String[] args) {
    InputStream is = null;
    try {
        is = new FileInputStream("/home/files/PropertyGroups.xml");
        String pgDef = IOUtils.toString(is);
        OKMPropertyGroup.getInstance().registerDefinition(null, pgDef);
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        IOUtils.closeQuietly(is);
    }
}
```

OKMRepository

On all methods you'll see parameter named "**token**". When accessing application across SOAP the login process returns a token, what is used to identify the user on all the exposed methods. From default application execution context you must use "**null**" value what indicates to the application must use the "**user session**".



On special cases you might be "**promoted as Administrator**" using the "**administrator token**".

```
String systemToken = DbSessionManager.getInstance().getSystemToken();
```

Methods

getRootFolder

Description:

Method	Return values	Description
getRootFolder(String token)	Folder	Returns the object Folder of node "/okm:root".

Example:

```
package com.openkm;

import com.openkm.api.OKMRepository;
import com.openkm.bean.Folder;

public class Test {
    public static void main(String[] args) {
        try {
            Folder fld = OKMRepository.getInstance().getRootFolder(null);
            System.out.println(fld);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getTrashFolder

Description:

Method	Return values	Description
getTrashFolder(String token)	Folder	Returns the object Folder of node "/okm:trash/{userId}".

The returned folder will be the user trash folder.



For example if the method is executed by the user "okmAdmin" then the folder returned will be "/okm:trash/okmAdmin".

Example:

```
package com.openkm;

import com.openkm.api.OKMRepository;
import com.openkm.bean.Folder;

public class Test {
    public static void main(String[] args) {
        try {
            Folder fld = OKMRepository.getInstance().getTrashFolder(null);
            System.out.println(fld);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getTrashFolderBase

Description:

Method	Return values	Description
getTrashFolderBase(String token)	Folder	Returns the object Folder of node "/okm:trash".

Example:

```
package com.openkm;

import com.openkm.api.OKMRepository;
import com.openkm.bean.Folder;

public class Test {
    public static void main(String[] args) {
        try {
            Folder fld = OKMRepository.getInstance().getTrashFolderBase(null);
            System.out.println(fld);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getTemplatesFolder

Description:

Method	Return values	Description
<i>getTemplatesFolder(String token)</i>	<i>Folder</i>	Returns the object <i>Folder</i> of node <i>"/okm:templates"</i> .

Example:

```
package com.openkm;

import com.openkm.api.OKMRepository;
import com.openkm.bean.Folder;

public class Test {
    public static void main(String[] args) {
        try {
            Folder fld = OKMRepository.getInstance().getTemplatesFolder(null);
            System.out.println(fld);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getPersonalFolder

Description:

Method	Return values	Description
<i>getPersonalFolder(String token)</i>	<i>Folder</i>	Returns the object <i>Folder</i> of node <i>"/okm:personal/{userId}"</i> .

The returned folder will be the user personal folder.



For example if the method is executed by the user "okmAdmin" then the folder returned will be *"/okm:personal/okmAdmin"*.

Example:

```
package com.openkm;

import com.openkm.api.OKMRepository;
import com.openkm.bean.Folder;

public class Test {
    public static void main(String[] args) {
        try {
            Folder fld = OKMRepository.getInstance().getPersonalFolder(null);
            System.out.println(fld);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
}
```

getPersonalFolderBase

Description:

Method	Return values	Description
<i>getPersonalFolderBase(String token)</i>	<i>Folder</i>	Returns the object Folder of node "/okm:personal".

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMRepository;  
import com.openkm.bean.Folder;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            Folder fld = OKMRepository.getInstance().getPersonalFolderBase(null);  
            System.out.println(fld);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

getMailFolder

Description:

Method	Return values	Description
<i>getMailFolder(String token)</i>	<i>Folder</i>	Returns the object Folder of node "/okm:mail/{userId}".

The returned folder will be the user mail folder.



For example if the method is executed by the user "okmAdmin" then the folder returned will be "/okm:mail/okmAdmin".

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMRepository;  
import com.openkm.bean.Folder;  
  
public class Test {
```

```
public static void main(String[] args) {
    try {
        Folder fld = OKMRepository.getInstance().getMailFolder(null);
        System.out.println(fld);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

getMailFolderBase

Description:

Method	Return values	Description
getMailFolderBase(String token)	Folder	Returns the object Folder of node "/okm:mail".

Example:

```
package com.openkm;

import com.openkm.api.OKMRepository;
import com.openkm.bean.Folder;

public class Test {
    public static void main(String[] args) {
        try {
            Folder fld = OKMRepository.getInstance().getMailFolderBase(null);
            System.out.println(fld);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getThesaurusFolder

Description:

Method	Return values	Description
getThesaurusFolder(String token)	Folder	Returns the object Folder of node "/okm:thesaurus".

Example:

```
package com.openkm;

import com.openkm.api.OKMRepository;
import com.openkm.bean.Folder;

public class Test {
    public static void main(String[] args) {
```

```

        try {
            Folder fld = OKMRepository.getInstance().getThesaurusFolder(null);
            System.out.println(fld);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

getCategoriesFolder

Description:

Method	Return values	Description
<i>getCategoriesFolder(String token)</i>	Folder	Returns the object Folder of node "/okm:categories".

Example:

```

package com.openkm;

import com.openkm.api.OKMRepository;
import com.openkm.bean.Folder;

public class Test {
    public static void main(String[] args) {
        try {
            Folder fld = OKMRepository.getInstance().getCategoriesFolder(null);
            System.out.println(fld);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

purgeTrash

Description:

Method	Return values	Description
<i>purgeTrash(String token)</i>	Folder	Definitively removes from repository all nodes to "/okm:trash/{userId}".

For example if the method is executed by the user "okmAdmin" then the purged trash will be "/okm:trash/okmAdmin".



When a node is purged it will only be able to be restored from a previously repository backup. The purge action removes the node definitely from the repository.

Example:


```
package com.openkm;

import com.openkm.api.OKMRepository;
import com.openkm.bean.Folder;

public class Test {
    public static void main(String[] args) {
        try {
            OKMRepository.getInstance().purgeTrash(null);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getUpdateMessage

Description:

Method	Return values	Description
getUpdateMessage(String token)	String	Retrieves a message when there is a new OpenKM release.
<p>There's an official OpenKM update message service available which is based on your local OpenKM version.</p> <div>  The most common message is that a new OpenKM version has been released. </div>		

Example:

```
package com.openkm;

import com.openkm.api.OKMRepository;

public class Test {
    public static void main(String[] args) {
        try {
            String msg = OKMRepository.getInstance().getUpdateMessage(null);
            System.out.println(msg);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getRepositoryUuid

Description:

Method	Return values	Description
--------	---------------	-------------

<i>getRepositoryUuid(String token)</i>	<i>String</i>	<i>Retrieves the installation unique identifier.</i>
---	----------------------	--

Example:

```
package com.openkm;

import com.openkm.api.OKMRepository;

public class Test {
    public static void main(String[] args) {
        try {
            String uuid = OKMRepository.getInstance().getRepositoryUuid(null);
            System.out.println(uuid);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

hasNode

Description:

Method	Return values	Description
<i>hasNode(String token, String nodeId)</i>	<i>boolean</i>	<i>Returns a node that indicate if a node exists or not.</i>
<i>The value of the parameter nodeId can be a valid UUID or path.</i>		

Example:

```
package com.openkm;

import com.openkm.api.OKMRepository;

public class Test {
    public static void main(String[] args) {
        try {
            boolean found = OKMRepository.getInstance().hasNode(null, "/okm:root/test")
            System.out.println(found);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getNodePath

Description:

Method	Return values	Description
---------------	----------------------	--------------------

<i>getNodePath(String token, String uuid)</i>	<i>String</i>	<i>Converts a node UUID to path.</i>
--	----------------------	--------------------------------------

Example:

```
package com.openkm;

import com.openkm.api.OKMRepository;

public class Test {
    public static void main(String[] args) {
        try {
            String path = OKMRepository.getInstance().getNodePath(null, "064ff51a-b81
            System.out.println(path);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getNodeUuid

Description:

<i>Method</i>	<i>Return values</i>	<i>Description</i>
<i>getNodeUuid(String token, String path)</i>	<i>String</i>	<i>Converts a node path to UUID.</i>

Example:

```
package com.openkm;

import com.openkm.api.OKMRepository;

public class Test {
    public static void main(String[] args) {
        try {
            String uuid = OKMRepository.getInstance().getNodeUuid(null, "/okm:root/tm
            System.out.println(uuid);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getAppVersion

Description:

<i>Method</i>	<i>Return values</i>	<i>Description</i>
<i>getAppVersion(String token)</i>	<i>AppVersion</i>	<i>Returns information about OpenKM version.</i>

Example:

```
package com.openkm;

import com.openkm.api.OKMRepository;
import com.openkm.bean.AppVersion;

public class Test {
    public static void main(String[] args) {
        try {
            AppVersion appVersion = OKMRepository.getInstance().getAppVersion(null);
            System.out.println(appVersion);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

copyAttributes

Description:

Method	Return values	Description
copyAttributes(String token, String srcId, String dstId, ExtendedAttributes extAttr)	void	Copy attributes from a node to other..

The values of the dstId parameter should be a node UUID or path.



- When the category parameter is true the original values of the categories will be copied.
- When the keywords parameter is true the original values of the keywords will be copied.
- When the property Groups parameter is true the original values of the metadata groups will be copied.
- When the notes parameter is true the original values of the notes will be copied.
- When the wiki parameter is true the original values of the wiki will be copied.

Example:

```
package com.openkm;

import com.openkm.api.OKMRepository;
import com.openkm.bean.ExtendedAttributes;

public class Test {
    public static void main(String[] args) {
```




```

        try {
            ExtendedAttributes extAttr = new ExtendedAttributes();
            extAttr.setKeywords(true);
            extAttr.setCategories(true);
            extAttr.setNotes(true);
            extAttr.setPropertyGroups(true);
            extAttr.setWiki(true);
            OKMRepository.getInstance().copyAttributes(null, "/okm:root/invoice.pdf",
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

registerPropertyGroups

Description:

Method	Return values	Description
registerPropertyGroups(String pgDefFile)	void	Apply the the XML groups definition into OpenKM.
<div>  <p>The method only can be executed by super user granted (ROLE_ADMIN member user).</p> <p>Take in mind chaging metadata is done in two steps, first modifying \$TOMCAT_HOME/PropertyGroups.xml file, take a look at registerDefinition in OKMPropertyGroup class for it. Then executing registerPropertyGroups to applying the changes.</p> </div>		

Example:

```

package com.openkm;

import com.openkm.core.Config;
import com.openkm.api.OKMRepository;

public class Test {
    public static void main(String[] args) {
        try {
            OKMRepository.getInstance().registerPropertyGroups(null, Config.PROPERTY_
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```


OKMSearch


Basics

Almosts all methods use `QueryParams`. Here there're some tips about how using it.

Variables	Type	Allow wildcards	Restrictions
domain	long	No.	<p>Available values:</p> <ul style="list-style-type: none"> <code>QueryParams.DOCUMENT</code> <code>QueryParams.FOLDER</code> <code>QueryParams.MAIL</code> <p>By default the value is set to <code>QueryParams.DOCUMENT</code>.</p> <p>For searching documents and folders use value:</p> <div>(<code>QueryParams.DOCUMENT</code> <code>QueryParams.FOLDER</code>)</div>
author	String	No.	Value must be a valid <code>userId</code> .
name	String	Yes.	
keywords	Set<String>	Yes.	
categories	Set<String>	No.	Values should be a category <code>UUID</code> , not use path value.
content		Yes.	

<i>mimeType</i>		No.	Value should be a valid and registered MIME type. Only can be applied to documents node.
<i>path</i>		No.	When empty is used by default "/okm:root" node. Value should be a valid path, not use a UUID value.
<i>lastModifiedFrom</i>	<i>Calendar</i>	No.	
<i>lastModifiedTo</i>	<i>Calendar</i>	No.	
<i>mailSubject</i>	<i>String</i>	Yes.	Only apply to mail nodes.
<i>mailFrom</i>	<i>String</i>	Yes.	Only apply to mail nodes.
<i>mailTo</i>		Yes.	Only apply to mail nodes.
<i>properties</i>	<i>Map<String,</i>	Yes on	On metadata field values like "date" can not be applied wilcards.

	String>	almost.	<p>The map of the properties is composed of pairs:</p> <p>('metadata_field_name','metada_field_value')</p> <p>For example:</p> <pre>Map<String, String> properties = new HashMap(); properties.put("okp:consulting.name", "name val</pre> <p>Filtering by range of dates:</p> <pre>Calendar to = Calendar.getInstance(); // today to.set(0, Calendar.HOUR); to.set(0, Calendar.MINUTE); to.set(0, Calendar.SECOND); to.set(0, Calendar.MILLISECOND); Calendar from = (Calendar) to.clone(); from.add(-3, Calendar.DATE); // three days befo Map<String,String> properties = new HashMap<>() properties.put("okp:consulting.date", ISO8601.f</pre> <div>  <p>When filtering by range of dates you must set both values (fr side.</p> </div> <p>To filtering by a metadata field of type multiple like this:</p> <pre><select label="Multiple" name="okp:consulting.m <option label="One" value="one"/> <option label="Two" value="two"/> <option label="Three" value="three" /> </select></pre> <p>You should do:</p> <pre>properties.put("okp:consulting.multiple", "one;</pre> <p>Where "one" and "two" are valid values and character ";" is used as sepa</p>
--	-------------------	---------	---



The search operation is done only by AND logic.

Wildcard examples:


Variable	Example	Description
name	test*.html	Any document that starts with characters "test" and ends with characters ".html"

name	test?.html	Any document that starts with characters "test" followed by a single character and ends with characters ".html"
name	?test*	Any of the documents where the first character doesn't matter, but is followed by the characters, "test".

Methods

findByContent

Description:

Method	Return values	Description
<i>findByContent(String content)</i>	<i>List<QueryResult></i>	Returns a list of results filtered by the value of the content parameter.
 The method only searches among all documents, it does not take in consideration any other kind of nodes.		

Example:

```
package com.openkm;

import com.openkm.api.OKMSearch;
import com.openkm.bean.QueryResult;

public class Test {
    public static void main(String[] args) {
        try {
            for (QueryResult qr : OKMSearch.getInstance().findByContent(null, "test"))
                System.out.println(qr);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

findByName

Description:

Method	Return values	Description

findByName(String token, String words)

List<QueryResult>

Returns a list of results filtered by the value of the name parameter.



The method only searches among all documents, it does not take in consideration any other kind of nodes.

Example:

```
package com.openkm;

import com.openkm.api.OKMSearch;
import com.openkm.bean.QueryResult;

public class Test {
    public static void main(String[] args) {
        try {
            for (QueryResult qr : OKMSearch.getInstance().findByName(null, "test*.htm")) {
                System.out.println(qr);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

findByKeywords

Description:

Method	Return values	Description
<i>findByKeywords(String token, Set<String> words)</i>	<i>List<QueryResult></i>	Returns a list of results filtered by the values of the keywords parameter.



The method only searches among all documents, it does not take in consideration any other kind of nodes.

Example:

```
package com.openkm;

import java.util.Arrays;
import java.util.HashSet;
import java.util.Set;

import com.openkm.api.OKMSearch;
import com.openkm.bean.QueryResult;
```

```

public class Test {
    public static void main(String[] args) {
        try {
            Set<String> keywords = new HashSet<>(Arrays.asList("test"));
            for (QueryResult qr : OKMSearch.getInstance().findByKeywords(null, keywords)) {
                System.out.println(qr);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

find

Description:

Method	Return values	Description
<i>find(String token, QueryParams params)</i>	<i>List<QueryResult></i>	Returns a list of results filtered by the values of the queryParams parameter.

Example:

```

package com.openkm;

import com.openkm.api.OKMSearch;
import com.openkm.bean.QueryResult;
import com.openkm.dao.bean.QueryParams;

public class Test {
    public static void main(String[] args) {
        try {
            QueryParams qParams = new QueryParams();
            qParams.setDomain(QueryParams.DOCUMENT);
            qParams.setName("test*.html");
            for (QueryResult qr : OKMSearch.getInstance().find(null, qParams)) {
                System.out.println(qr);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

findPaginated

Description:

Method	Return values	Description
<i>findPaginated(String token, QueryParams params, int offset, int limit)</i>	<i>ResultSet</i>	Returns a list of paginated results filtered by the values of the queryParams parameter.



The parameter "limit" and "offset" allows you to retrieve just a portion of the results of a query.

- The parameter "limit" is used to limit the number of results returned.
- The parameter "offset" says to skip that many results before the beginning to return results.



For example if your query has 1000 results, but you only want to return the first 10, you should use these values:

- limit=10
- offset=0

Now suppose you want to show the results from 11-20, you should use these values:

- limit=10
- offset=10

Example:

```
package com.openkm;

import com.openkm.api.OKMSearch;
import com.openkm.bean.QueryResult;
import com.openkm.bean.ResultSet;
import com.openkm.dao.bean.QueryParams;

public class Test {
    public static void main(String[] args) {
        try {
            QueryParams qParams = new QueryParams();
            qParams.setDomain(QueryParams.DOCUMENT);
            qParams.setName("test*.html");
            ResultSet rs = OKMSearch.getInstance().findPaginated(null, qParams, 20, 1);
            System.out.println("Total results:"+rs.getTotal());

            for (QueryResult qr : rs.getResults()) {
                System.out.println(qr);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

saveSearch

Description:

Method	Return values	Description

<code>saveSearch(String token, QueryParams params)</code>	Long	Saves a search parameters.
The variable <code>queryName</code> of the parameter <code>params</code> , should have to be initialized.		


```
package com.openkm;

import com.openkm.api.OKMSearch;
import com.openkm.dao.bean.QueryParams;

public class Test {
    public static void main(String[] args) {
        try {
            QueryParams qParams = new QueryParams();
            qParams.setDomain(QueryParams.DOCUMENT);
            qParams.setName("test*.html");
            // Save the search to be used later
            qParams.setQueryName("sample search");
            OKMSearch.getInstance().saveSearch(null, qParams);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

updateSearch

Description:

Method	Return values	Description
<code>updateSearch(String token, QueryParams params)</code>	void	Saves a search parameters.
 Only can be updated as a saved search created by the same user who's executing the method.		

```
package com.openkm;


import com.openkm.api.OKMSearch;
import com.openkm.dao.bean.QueryParams;

public class Test {
    public static void main(String[] args) {
        try {
            for (QueryParams qParams : OKMSearch.getInstance().getAllSearchs(null)) {
                if (qParams.getQueryName().equals("sample search")) {
                    // Change some value.
                    qParams.setName("admin*.html");
                    OKMSearch.getInstance().updateSearch(null, qParams);
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
}
```

getSearch


Description:

Method	Return values	Description
getSearch(String token, int qpId)	QueryParams	Gets saved search parameters.
<div> Only can be updated as a saved search created by the same user user who's executing the method.</div>		

```
package com.openkm;  
  
import com.openkm.api.OKMSearch;  
import com.openkm.dao.bean.QueryParams;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            int qpId = 12; // Some valid search id  
            QueryParams qParams = OKMSearch.getInstance().getSearch(null, qpId);  
            System.out.println(qParams);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

getAllSearchs

Description:

Method	Return values	Description
getAllSearchs(String token)	List<QueryParams>	Retrieves a list of all saved search parameters.
<div> Only can be updated as a saved search created by the same user user who's executing the method.</div>		

```
package com.openkm;  
  
import com.openkm.api.OKMSearch;  
import com.openkm.dao.bean.QueryParams;  
  
public class Test {  
    public static void main(String[] args) {
```


```

        try {
            for (QueryParams qParams : OKMSearch.getInstance().getAllSearchs(null)) {
                System.out.println(qParams);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

deleteSearch

Description:

Method	Return values	Description
deleteSearch(String token, long qpId)	void	Deletes a saved search parameters.
 Only can be updated as a saved search created by the same user user who's executing the method.		

```

package com.openkm;


import com.openkm.api.OKMSearch;

public class Test {
    public static void main(String[] args) {
        try {
            int qpId = 12; // Some valid search id
            OKMSearch.getInstance().deleteSearch(null, qpId);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

getKeywordMap

Description:

Method	Return values	Description
getKeywordMap(String token, List<String> filter)	Map<String, Integer>	Returns a map of keywords with its count value filtered by other keywords.
 Example: <ul style="list-style-type: none"> • Doc1.txt has keywords "test", "one", "two". • Doc2.txt has keywords "test", "one" 		

- Doc3.txt has keywords "test", "three".

The results filtering by "test" -> "one", "two", "three".

The results filtering by "one" -> "test", "two".

The results filtering by "two" -> "test", "one".

The results filtering by "three" -> "test".

The results filtering by "one" and "two" -> "test"

```
package com.openkm;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Map;

import com.openkm.api.OKMSearch;

public class Test {
    public static void main(String[] args) {
        try {
            // All keywords without filtering
            System.out.println("Without filtering");
            Map<String, Integer> keywords = OKMSearch.getInstance().getKeywordMap(null);

            for (String key : keywords.keySet()) {
                System.out.println(key + " is used : " + keywords.get(key) );
            }

            // Keywords filtered
            System.out.println("Filtering");
            keywords = OKMSearch.getInstance().getKeywordMap(null, Arrays.asList("test", "three"));

            for (String key : keywords.keySet()) {
                System.out.println(key + " is used : " + keywords.get(key) );
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getCategorizedDocuments

Description:

Method	Return values	Description
getCategorizedDocuments(String token, String categoryId)	List<Document>	Retrieves a list of all documents related with a category.
The values of the categoryId parameter should be a category folder UUID or path.		

```

package com.openkm;

import com.openkm.api.OKMSearch;
import com.openkm.bean.Document;

public class Test {
    public static void main(String[] args) {
        try {
            for (Document doc : OKMSearch.getInstance().getCategorizedDocuments(null,
                System.out.println(doc);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

getCategorizedFolders

Description:

Method	Return values	Description
<i>getCategorizedFolders(String token, String categoryId)</i>	<i>List<Folder></i>	Retrieves a list of all folders related with a category.
The values of the categoryId parameter should be a category folder UUID or path.		

```

package com.openkm;

import com.openkm.api.OKMSearch;
import com.openkm.bean.Folder;

public class Test {
    public static void main(String[] args) {
        try {
            for (Folder fld : OKMSearch.getInstance().getCategorizedFolders(null, "/o.
                System.out.println(fld);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

getCategorizedMails

Description:

Method	Return values	Description
<i>getCategorizedMails(String token, String</i>	<i>List<Mail></i>	Retrieves a list of all mails related with a

categoryId)

category.

The values of the categoryId parameter should be a category folder UUID or path.

```
package com.openkm;

import com.openkm.api.OKMSearch;
import com.openkm.bean.Mail;

public class Test {
    public static void main(String[] args) {
        try {
            for (Mail mail : OKMSearch.getInstance().getCategorizedMails(null, "/okm:")) {
                System.out.println(mail);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getDocumentsByKeyword

Description:

Method	Return values	Description
getDocumentsByKeyword(String token, String keyword)	List<Document>	Retrieves a list of all documents related with a keyword.

```
package com.openkm;

import com.openkm.api.OKMSearch;
import com.openkm.bean.Document;

public class Test {
    public static void main(String[] args) {
        try {
            for (Document doc : OKMSearch.getInstance().getDocumentsByKeyword(null, "")) {
                System.out.println(doc);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getFoldersByKeyword

Description:

Method	Return values	Description

getFoldersByKeyword(String token, String keyword)

List<Folder>

Retrieves a list of all folders related with a keyword.

```
package com.openkm;

import com.openkm.api.OKMSearch;
import com.openkm.bean.Folder;

public class Test {
    public static void main(String[] args) {
        try {
            for (Folder fld : OKMSearch.getInstance().getFoldersByKeyword(null, "test")) {
                System.out.println(fld);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getMailsByKeyword

Description:

<i>Method</i>	<i>Return values</i>	<i>Description</i>
<i>getMailsByKeyword(String token, String keyword)</i>	<i>List<Mail></i>	<i>Retrieves a list of all mails related with a keyword.</i>

```
package com.openkm;

import com.openkm.api.OKMSearch;
import com.openkm.bean.Mail;

public class Test {
    public static void main(String[] args) {
        try {
            for (Mail mail : OKMSearch.getInstance().getMailsByKeyword(null, "test")) {
                System.out.println(mail);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getDocumentsByPropertyValue

Description:

<i>Method</i>	<i>Return values</i>	<i>Description</i>

<i>getDocumentsByPropertyValue(String token, String group, String property, String value</i>	<i>List<Document></i>	<i>Retrieves a list of all documents related with a metadata field value.</i>
---	------------------------------------	---

```
package com.openkm;

import com.openkm.api.OKMSearch;
import com.openkm.bean.Document;

public class Test {
    public static void main(String[] args) {
        try {
            for (Document doc : OKMSearch.getInstance().getDocumentsByPropertyValue(token, group, property, value)) {
                System.out.println(doc);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getFoldersByPropertyValue

Description:

Method	Return values	Description
<i>getFoldersByPropertyValue(String token, String group, String property, String value</i>	<i>List<Folder></i>	<i>Retrieves a list of all folders related with a metadata field value.</i>

```
package com.openkm;

import com.openkm.api.OKMSearch;
import com.openkm.bean.Folder;

public class Test {
    public static void main(String[] args) {
        try {
            for (Folder fld : OKMSearch.getInstance().getFoldersByPropertyValue(token, group, property, value)) {
                System.out.println(fld);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getMailsByPropertyValue

Description:

Method	Return values	Description

getMailsByPropertyValue(String token, String group, String property, String value

List<Mail>

Retrieves a list of all mails related with a metadata field value.

```
package com.openkm;

import com.openkm.api.OKMSearch;
import com.openkm.bean.Mail;

public class Test {
    public static void main(String[] args) {
        try {
            for (Mail mail : OKMSearch.getInstance().getMailsByPropertyValue(null, "o", "o", "o")) {
                System.out.println(mail);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

findSimpleQuery

Description:

Method	Return values	Description
<i>findSimpleQuery(String token, String statement)</i>	<i>List<QueryResult></i>	<i>Returns a list of results filtered by the values of the statement parameter.</i>



The **syntax** to use in the statement parameter is the pair '**field:value**'. For example:

- "name:grial" is filtering field name by word grial.

More information about lucene sintaxis at [Lucene query syntax](#).

```
package com.openkm;

import com.openkm.api.OKMSearch;
import com.openkm.bean.QueryResult;

public class Test {
    public static void main(String[] args) {
        try {
            for (QueryResult qr : OKMSearch.getInstance().findSimpleQuery(null, "name:grial")) {
                System.out.println(qr);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

findSimpleQueryPaginated

Description:

Method	Return values	Description
<i>findSimpleQueryPaginated(String token, String statement, int offset, int limit)</i>	<i>ResultSet</i>	Returns a list of paginated results filtered by the values of the statement parameter.

The **syntax** to use in the statement parameter is the pair '**field:value**'. For example:

- "name:grial" is filtering field name by word grial.

More information about lucene sintaxis at [Lucene query syntax](#).

✓ The parameter "limit" and "offset" allows you to retrieve just a portion of the results of a query.

- The parameter "limit" is used to limit the number of results returned.
- The parameter "offset" says to skip that many results before the beginning to return results.

i For example if your query has 1000 results, but you only want to return the first 10, you should use these values:

- limit=10
- offset=0

Now suppose you want to show the results from 11-20, you should use these values:

- limit=10
- offset=10

```

package com.openkm;

import com.openkm.api.OKMSearch;
import com.openkm.bean.QueryResult;
import com.openkm.bean.ResultSet;

public class Test {
    public static void main(String[] args) {
        try {
            ResultSet rs = OKMSearch.getInstance().findSimpleQueryPaginated(null, "na
            System.out.println("Total results:"+rs.getTotal());


            for (QueryResult qr : rs.getResults()) {
                System.out.println(qr);
            }
        }
    }
}

```

```
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

findMoreLikeThis

Description:

Method	Return values	Description
<i>findMoreLikeThis(String token, String uuid, int maxResults)</i>	<i>ResultSet</i>	Returns a list of documents that are considered similar by search engine.
<p>The uuid is a document UUID.</p> <p>The max value is used to limit the number of results returned.</p> <div> The method can only be used with documents.</div>		

```
package com.openkm;  
  
import com.openkm.api.OKMSearch;  
import com.openkm.bean.QueryResult;  
import com.openkm.bean.ResultSet;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            ResultSet rs = OKMSearch.getInstance().findMoreLikeThis(null, "f123a950-0  
            System.out.println("Total results:"+rs.getTotal());  
  
            for (QueryResult qr : rs.getResults()) {  
                System.out.println(qr);  
            }  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

OKMStats

Basics

On all methods you'll see parameter named "**token**". When accessing application across SOAP the login process returns a token, what is used to identify the user on all the exposed methods. From default application execution context you must use "**null**" value what indicates to the application must use the "**user session**".



On special cases you might be "**promoted as Administrator**" using the "**administrator token**".

```
String systemToken = DbSessionManager.getInstance().getSystemToken();
```

Methods

getDocumentsByContext

Description:

Method	Return values	Description
getDocumentsByContext(String token)	StatsInfo	Return stats information about documents by context.

Example:

```
package com.openkm;

import com.openkm.api.OKMStats;
import com.openkm.bean.StatsInfo;

public class Test {
    public static void main(String[] args) {
        try {
            StatsInfo sInfo = OKMStats.getInstance().getDocumentsByContext(null);
            System.out.println(sInfo);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getDocumentsSizeByContext

Description:

Method	Return values	Description
getDocumentsByContext(String token)	StatsInfo	Return stats information about documents size by context.

Example:

```
package com.openkm;

import com.openkm.api.OKMStats;
import com.openkm.bean.StatsInfo;

public class Test {
    public static void main(String[] args) {
        try {
            StatsInfo sInfo = OKMStats.getInstance().getDocumentsSizeByContext(null);
            System.out.println(sInfo);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getFoldersByContext

Description:

Method	Return values	Description
getFoldersByContext(String token)	StatsInfo	Return stats information about folders by context.

Example:

```
package com.openkm;

import com.openkm.api.OKMStats;
import com.openkm.bean.StatsInfo;

public class Test {
    public static void main(String[] args) {
        try {
            StatsInfo sInfo = OKMStats.getInstance().getFoldersByContext(null);
            System.out.println(sInfo);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getMailsByContext

Description:

Method	Return values	Description
getMailsByContext(String token)	StatsInfo	Return stats information about mails by context.

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMStats;  
import com.openkm.bean.StatsInfo;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            StatsInfo sInfo = OKMStats.getInstance().getMailsByContext(null);  
            System.out.println(sInfo);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

OKMUserConfig

On all methods you'll see parameter named "**token**". When accessing application across SOAP the login process returns a token, what is used to identify the user on all the exposed methods. From default application execution context you must use "**null**" value what indicates to the application must use the "**user session**".



On special cases you might be "**promoted as Administrator**" using the "**administrator token**".


```
String systemToken = DbSessionManager.getInstance().getSystemToken();
```

Methods

setHome

Description:

Method	Return values	Description
setHome(String token, String path)	void	Sets the user home.

 By default user home value is "/okm:root".

Example:

```
package com.openkm;

import com.openkm.api.OKMUserConfig;

public class Test {
    public static void main(String[] args) {
        try {
            OKMUserConfig.getInstance().setHome(null, "/okm:root/test");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getConfig

Description:

Method	Return values	Description

<i>getConfig(String token)</i>	<i>UserConfig</i>	<i>Return the user configuration.</i>
<p><i>Returned values are:</i></p> <ul style="list-style-type: none">• <i>The userId.</i>• <i>The home path.</i>• <i>The home type (folder, document, or mail).</i>• <i>The home node (the UUID).</i>		

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMUserConfig;  
import com.openkm.dao.bean.UserConfig;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            UserConfig uc = OKMUserConfig.getInstance().getConfig(null);  
            System.out.println(uc);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```


OKMWorkflow

On all methods you'll see parameter named "**token**". When accessing application across SOAP the login process returns a token, what is used to identify the user on all the exposed methods. From default application execution context you must use "**null**" value what indicates to the application must use the "**user session**".



On special cases you might be "**promoted as Administrator**" using the "**administrator token**".

```
String systemToken = DbSessionManager.getInstance().getSystemToken();
```

Methods

registerProcessDefinition

Description:

Method	Return values	Description
registerProcessDefinition(String token, InputStream is)	void	Registers a new workflow.

Example:

```
package com.openkm;

import java.io.FileInputStream;
import java.io.InputStream;

import org.apache.commons.io.IOUtils;

import com.openkm.api.OKMWorkflow;

public class Test {
    public static void main(String[] args) {
        try {
            InputStream is = new FileInputStream("/opt/files/Purchase.par");
            OKMWorkflow.getInstance().registerProcessDefinition(null, is);
            IOUtils.closeQuietly(is);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

deleteProcessDefinition

Description:

Method	Return values	Description

<code>deleteProcessDefinition(String token, long processDefinitionId)</code>	<code>void</code>	<i>Deletes a workflow.</i>
<i>The parameter processDefinitionId value is a valid workflow process definition.</i>		

Example:

```
package com.openkm;

import com.openkm.api.OKMWorkflow;

public class Test {
    public static void main(String[] args) {
        try {
            long pdId = 5; // Valid workflow process definition
            OKMWorkflow.getInstance().deleteProcessDefinition(null, pdId);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

`getProcessDefinitionImage`

Description:

Method	Return values	Description
<code>getProcessDefinitionImage(String token, long processDefinitionId, String node)</code>	<code>byte[]</code>	<i>Returns a workflow image process definition.</i>
<i>The parameter processDefinitionId value is a valid workflow process definition.</i>		
<i>The parameter node is a node with a workflow in execution.</i>		

Example:

```
package com.openkm;

import java.io.ByteArrayInputStream;
import java.io.File;
import java.io.FileOutputStream;

import org.apache.commons.io.IOUtils;

import com.openkm.api.OKMWorkflow;

public class Test {
    public static void main(String[] args) {
        try {
            long pdId = 5; // Valid workflow process definition
```

```

        byte[] img = OKMWorkflow.getInstance().getProcessDefinitionImage(null, pdId);
        ByteArrayInputStream in = new ByteArrayInputStream(img);
        FileOutputStream out = new FileOutputStream(new File("/home/test/img.jpg"));
        IOUtils.copy(in, out);
        IOUtils.closeQuietly(in);
        IOUtils.closeQuietly(out);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

getProcessDefinitionForms

Description:

Method	Return values	Description
Map<String, List<FormElement>> getProcessDefinitionForms(String token, long processDefinitionId)	Map<String, List<FormElement>>	Returns a map with all workflow form definitions.
The parameter processDefinitionId value is a valid workflow process definition.		

Example:

```

package com.openkm;

import java.util.List;
import java.util.Map;

import com.openkm.api.OKMWorkflow;
import com.openkm.bean.form.FormElement;

public class Test {
    public static void main(String[] args) {
        try {
            long pdId = 5; // Valid workflow process definition
            Map<String, List<FormElement>> map = OKMWorkflow.getInstance().getProcessDefinitionForms(token, pdId);
            for (String key : map.keySet()) {
                System.out.println("group:" + key);
                for (FormElement fe : map.get(key)) {
                    System.out.println(fe);
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

runProcessDefinition

Description:

Method	Return values	Description
<i>runProcessDefinition(String token, long processDefinitionId, String uuid, List<FormElement> variables)</i>	<i>ProcessInstance</i>	Executes a workflow on a node.
<p>The parameter processDefinitionId value is a valid workflow process definition.</p> <p>The parameter uuid can be any document, mail, or folder UUID.</p> <p>The parameter values are form element values needed for starting the workflow (not all workflows need form values for starting).</p>		

Example:

```
package com.openkm;

import java.util.ArrayList;
import java.util.List;

import com.openkm.api.OKMWorkflow;
import com.openkm.bean.form.FormElement;
import com.openkm.bean.form.Input;
import com.openkm.bean.form.TextArea;

public class Test {
    public static void main(String[] args) {
        try {
            long pdId = 8041; // Some valid workflow process definition id
            List<FormElement> feList = new ArrayList<>(); // Case as part of starting
            Input price = new Input();
            price.setName("price");
            price.setValue("1000");
            feList.add(price);
            TextArea textArea = new TextArea();
            textArea.setName("description");
            textArea.setValue("some description here");
            feList.add(textArea);
            OKMWorkflow.getInstance().runProcessDefinition(null, pdId, "f86cc22d-9b50
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

sendProcessInstanceSignal

Description:

Method	Return values	Description
<i>sendProcessInstanceSignal(String token, long processInstanceId,</i>		Send a signal to a

<i>String transitionName)</i>	<i>ProcessInstance</i>	<i>process intance id.</i>
<p>The parameter <i>processInstanceId</i> value is a valid workflow process instance.</p> <p>The parameter <i>uuid</i> can be any document, mail, or folder UUID.</p> <p>The parameter <i>transactionName</i> is the choosen transaction.</p>		

Example:

```
package com.openkm;

import com.openkm.api.OKMWorkflow;

public class Test {
    public static void main(String[] args) {
        try {
            long pId = 1046; // Some valid process instance id
            OKMWorkflow.getInstance().sendProcessInstanceSignal(null, pId, "jump2");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

endProcessInstance

Description:

<i>Method</i>	<i>Return values</i>	<i>Description</i>
<i>endProcessInstance(String token, long processInstanceId)</i>	<i>void</i>	<i>Ends a process intance id.</i>
<p>The parameter <i>processInstanceId</i> value is a valid workflow process instance.</p>		

Example:

```
package com.openkm;

import com.openkm.api.OKMWorkflow;

public class Test {
    public static void main(String[] args) {
        try {
            long pId = 1046; // Some valid process instance id
            OKMWorkflow.getInstance().endProcessInstance(null, pId);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

deleteProcessInstance

Description:

Method	Return values	Description
<i>deleteProcessInstance(String token, long processInstanceId)</i>	<i>void</i>	<i>Delete a process instance id.</i>
<i>The parameter processInstanceId value is a valid workflow process instance.</i>		

Example:

```
package com.openkm;

import com.openkm.api.OKMWorkflow;

public class Test {
    public static void main(String[] args) {
        try {
            long pId = 1046; // Some valid process instance id
            OKMWorkflow.getInstance().deleteProcessInstance(null, pId);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

findProcessInstances

Description:

Method	Return values	Description
<i>findProcessInstances(String token, long processDefinitionId)</i>	<i>List<ProcessInstance></i>	<i>Retrieves a list of all process instances of some registered workflows definition.</i>
<i>The parameter processDefinitionId value is a valid workflow process definition.</i>		

Example:

```
package com.openkm;

import com.openkm.api.OKMWorkflow;
import com.openkm.bean.workflow.ProcessDefinition;
import com.openkm.bean.workflow.ProcessInstance;

public class Test {
    public static void main(String[] args) {
```

```

        try {
            // Get all workflow definitions
            for (ProcessDefinition pd : OKMWorkflow.getInstance().findAllProcessDefin
                System.out.println("WF definition: "+pd);

            // Get all process of some workflow definition
            for (ProcessInstance pi : OKMWorkflow.getInstance().findProcessInstan
                System.out.println("PI: "+pi);
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

findAllProcessDefinitions

Description:

Method	Return values	Description
<i>findAllProcessDefinitions(String token, long processDefinitionId)</i>	<i>List<ProcessDefinition></i>	Retrieves a list of all registered workflows definitions.

Example:

```

package com.openkm;

import com.openkm.api.OKMWorkflow;
import com.openkm.bean.workflow.ProcessDefinition;

public class Test {
    public static void main(String[] args) {
        try {
            // Get all workflow definitions
            for (ProcessDefinition pd : OKMWorkflow.getInstance().findAllProcessDefin
                System.out.println("WF definition: "+pd);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

findLatestProcessDefinitions

Description:

Method	Return values	Description
<i>findLatestProcessDefinitions(String token)</i>	<i>List<ProcessDefinition></i>	Retrieves a list of the last workflows definitions.



Several versions of the same workflow can be registered.

Example:


```
package com.openkm;

import com.openkm.api.OKMWorkflow;
import com.openkm.bean.workflow.ProcessDefinition;

public class Test {
    public static void main(String[] args) {
        try {
            // Get all workflow definitions
            for (ProcessDefinition pd : OKMWorkflow.getInstance().findLatestProcessDefinitions()) {
                System.out.println("WF definition: "+pd);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

findLastProcessDefinition

Description:

Method	Return values	Description
<i>findLastProcessDefinition(String token, String name)</i>	<i>ProcessDefinition</i>	Retrieves last workflow definition of some specific workflow.
The parameter name identifies an specific workflow definitions group.		
 Several workflow definition versions that have the same name.		

Example:

```
package com.openkm;

import com.openkm.api.OKMWorkflow;
import com.openkm.bean.workflow.ProcessDefinition;

public class Test {
    public static void main(String[] args) {
        try {
            ProcessDefinition pd = OKMWorkflow.getInstance().findLastProcessDefinition(token, name);
            System.out.println("WF definition: " + pd);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```




```

    }
}
}

```

findAllProcessDefinitionVersions

Description:

Method	Return values	Description
<i>findAllProcessDefinitionVersions(String token, String name)</i>	<i>ProcessDefinition</i>	Retrieves all workflow definition versions of some specific workflow.
The parameter name identifies an specific workflow definitions group.		
 Several workflow definition versions that have the same name.		

Example:

```

package com.openkm;

import com.openkm.api.OKMWorkflow;
import com.openkm.bean.workflow.ProcessDefinition;

public class Test {
    public static void main(String[] args) {
        try {
            for (ProcessDefinition pd : OKMWorkflow.getInstance().findAllProcessDefinitionVersions("WF definition: " + pd);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}

```

getProcessInstance

Description:

Method	Return values	Description
<i>getProcessInstance(String token, long processInstanceId)</i>	<i>ProcessInstance</i>	Returns the process instance.
The parameter processInstanceId is a valid process instance id.		

Example:

```

package com.openkm;

import com.openkm.api.OKMWorkflow;
import com.openkm.bean.workflow.ProcessInstance;

public class Test {
    public static void main(String[] args) {
        try {
            long piId = 8108; // Some valid process instance id
            ProcessInstance pi = OKMWorkflow.getInstance().getProcessInstance(null, piId);
            System.out.println("PI: " + pi);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

suspendProcessInstance

Description:

Method	Return values	Description
suspendProcessInstance(String token, long processInstanceId)	void	Suspends the process instance.
The parameter processInstanceId is a valid process instance id.		

Example:

```

package com.openkm;

import com.openkm.api.OKMWorkflow;
import com.openkm.bean.workflow.ProcessInstance;

public class Test {
    public static void main(String[] args) {
        try {
            long piId = 8108; // Some valid process instance id
            OKMWorkflow.getInstance().suspendProcessInstance(null, piId);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

resumeProcessInstance

Description:

Method	Return values	Description
resumeProcessInstance(String token, long processInstanceId)	void	Resume the process instance.

The parameter `processInstanceId` is a valid process instance id.

Example:

```
package com.openkm;

import com.openkm.api.OKMWorkflow;
import com.openkm.bean.workflow.ProcessInstance;

public class Test {
    public static void main(String[] args) {
        try {
            long piId = 8108; // Some valid process instance id
            OKMWorkflow.getInstance().resumeProcessInstance(null, piId);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

addProcessInstanceVariable

Description:

Method	Return values	Description
<code>addProcessInstanceVariable(String token, long processInstanceId, String name, Object value)</code>	void	Add a variable in a process instance.
The parameter <code>processInstanceId</code> is a valid process instance id.		

Example:

```
package com.openkm;

import com.openkm.api.OKMWorkflow;
import com.openkm.bean.workflow.ProcessInstance;

public class Test {
    public static void main(String[] args) {
        try {
            long piId = 8108; // Some valid process instance id
            OKMWorkflow.getInstance().addProcessInstanceVariable(null, piId, "key", ""
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

deleteProcessInstanceVariable

Description:

Method	Return values	Description
<i>deleteProcessInstanceVariable(String token, long processInstanceId, String name)</i>	void	Delete a variable in a process instance.
The parameter processInstanceId is a valid process instance id.		

Example:

```
package com.openkm;

import com.openkm.api.OKMWorkflow;
import com.openkm.bean.workflow.ProcessInstance;

public class Test {
    public static void main(String[] args) {
        try {
            long piId = 8108; // Some valid process instance id
            OKMWorkflow.getInstance().deleteProcessInstanceVariable(null, piId, "key")
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

findUserTaskInstances

Description:

Method	Return values	Description
<i>findUserTaskInstances(String token)</i>	List<TaskInstance>	Retrieves a list of task instances assigned to the user.

Example:

```
package com.openkm;

import com.openkm.api.OKMWorkflow;
import com.openkm.bean.workflow.TaskInstance;

public class Test {
    public static void main(String[] args) {
        try {
            // Get all user task instances
            for (TaskInstance ti : OKMWorkflow.getInstance().findUserTaskInstances(null, null)) {
                System.out.println(ti);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

findPooledTaskInstances

Description:

Method	Return values	Description
<i>findPooledTaskInstances(String token)</i>	<i>List<TaskInstance></i>	Retrieves a list of pooled task instances assigned to the user.

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMWorkflow;  
import com.openkm.bean.workflow.TaskInstance;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            // Get all user task instances  
            for (TaskInstance ti : OKMWorkflow.getInstance().findPooledTaskInstances(.  
                System.out.println(ti);  
            }  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

findTaskInstances

Description:

Method	Return values	Description
<i>findTaskInstances(String token, long processInstanceId)</i>	<i>List<TaskInstance></i>	Retrieves a list of task instances by process instance.
The parameter processInstanceId is a valid process instance id.		

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMWorkflow;  
import com.openkm.bean.workflow.TaskInstance;  
  
public class Test {  
    public static void main(String[] args) {  
        try {
```

```

        // Get all user task instances
        long piId = 8108; // Some valid process instance id
        for (TaskInstance ti : OKMWorkflow.getInstance().findTaskInstances(null, piId)) {
            System.out.println(ti);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

setTaskInstanceValues

Description:

Method	Return values	Description
setTaskInstanceValues(String token, long taskInstanceId, String transitionName, List<FormElement> values)	void	Set a task instance values.
<p>The parameter taskInstanceId is a valid task instance id.</p> <p>The parameter transitionName is the choosen transaction.</p> <p>The parameter values are form element values needed for starting the workflow (not all workflow tasks need form values).</p>		

Example:

```

package com.openkm;

import java.util.ArrayList;
import java.util.List;

import com.openkm.api.OKMWorkflow;
import com.openkm.bean.form.FormElement;

public class Test {
    public static void main(String[] args) {
        try {
            long tiId = 8110; // Some valid task instance id
            List<FormElement> feList = new ArrayList<>();
            OKMWorkflow.getInstance().setTaskInstanceValues(null, tiId, "approve", feList);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

addTaskInstanceComment

Description:

--

Method	Return values	Description
<i>addTaskInstanceComment(String token, long taskInstanceId, String message)</i>	<i>void</i>	Set a task instance comment.
The parameter <i>taskInstanceId</i> is a valid task instance id.		

Example:

```
package com.openkm;

import java.util.ArrayList;
import java.util.List;

import com.openkm.api.OKMWorkflow;

public class Test {
    public static void main(String[] args) {
        try {
            long tiId = 8110; // Some valid task instance id
            OKMWorkflow.getInstance().addTaskInstanceComment(null, tiId, "here the co.
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getTaskInstance

Description:

Method	Return values	Description
<i>getTaskInstance(String token, long taskInstanceId)</i>	<i>TaskInstance</i>	Returns a task instance.
The parameter <i>taskInstanceId</i> is a valid task instance id.		

Example:

```
package com.openkm;

import com.openkm.api.OKMWorkflow;
import com.openkm.bean.workflow.TaskInstance;

public class Test {
    public static void main(String[] args) {
        try {
            long tiId = 8110; // Some valid task instance id
            TaskInstance ti = OKMWorkflow.getInstance().getTaskInstance(null, tiId);
            System.out.println(ti);
        } catch (Exception e) {
        }
    }
}
```

```
        e.printStackTrace();
    }
}
```

setTaskInstanceActorId

Description:

Method	Return values	Description
setTaskInstanceActorId(String token, long taskInstanceId, String actorId)	void	Set the actor to a task instance.
The parameter taskInstanceId is a valid task instance id.		
The parameter actorId must be some valid OpenKM userId.		

Example:

```
package com.openkm;

import com.openkm.api.OKMWorkflow;

public class Test {
    public static void main(String[] args) {
        try {
            long tiId = 8110; // Some valid task instance id
            OKMWorkflow.getInstance().setTaskInstanceActorId(null, tiId, "okmAdmin");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

addTaskInstanceVariable

Description:

Method	Return values	Description
addTaskInstanceVariable(String token, long taskInstanceId, String name, Object value)	void	Add variable into a task instance.
The parameter taskInstanceId is a valid task instance id.		

Example:

```
package com.openkm;

import com.openkm.api.OKMWorkflow;

public class Test {
    public static void main(String[] args) {
        try {
            long tiId = 8110; // Some valid task instance id
            OKMWorkflow.getInstance().addTaskInstanceVariable(null, tiId, "key", "val");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

deleteTaskInstanceVariable

Description:

Method	Return values	Description
deleteTaskInstanceVariable(String token, long taskInstanceId, String name)	void	Delete variable from a task instance.
The parameter taskInstanceId is a valid task instance id.		

Example:

```
package com.openkm;

import com.openkm.api.OKMWorkflow;

public class Test {
    public static void main(String[] args) {
        try {
            long tiId = 8110; // Some valid task instance id
            OKMWorkflow.getInstance().deleteTaskInstanceVariable(null, tiId, "key");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

startTaskInstance

Description:

Method	Return values	Description

<i>startTaskInstance(String token, long taskInstanceId)</i>	<i>void</i>	<i>Starts a task instance.</i>
<i>The parameter taskInstanceId is a valid task instance id.</i>		

Example:

```
package com.openkm;

import com.openkm.api.OKMWorkflow;

public class Test {
    public static void main(String[] args) {
        try {
            long tiId = 8110; // Some valid task instance id
            OKMWorkflow.getInstance().startTaskInstance(null, tiId);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

endTaskInstance

Description:

<i>Method</i>	<i>Return values</i>	<i>Description</i>
<i>endTaskInstance(String token, long taskInstanceId, String transitionName)</i>	<i>void</i>	<i>Ends a task instance.</i>
<i>The parameter taskInstanceId is a valid task instance id.</i>		
<i>The parameter transitionName is a transaction name.</i>		

Example:

```
package com.openkm;

import com.openkm.api.OKMWorkflow;

public class Test {
    public static void main(String[] args) {
        try {
            long tiId = 8110; // Some valid task instance id
            OKMWorkflow.getInstance().endTaskInstance(null, tiId, "end");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

suspendTaskInstance

Description:

Method	Return values	Description
<i>suspendTaskInstance(String token, long taskInstanceId)</i>	<i>void</i>	<i>Suspend a task instance.</i>
<i>The parameter taskInstanceId is a valid task instance id.</i>		

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMWorkflow;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            long tiId = 8110; // Some valid task instance id  
            OKMWorkflow.getInstance().suspendTaskInstance(null, tiId);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

resumeTaskInstance

Description:

Method	Return values	Description
<i>resumeTaskInstance(String token, long taskInstanceId)</i>	<i>void</i>	<i>Resume a task instance.</i>
<i>The parameter taskInstanceId is a valid task instance id.</i>		

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMWorkflow;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            long tiId = 8110; // Some valid task instance id  
            OKMWorkflow.getInstance().resumeTaskInstance(null, tiId);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
}
```

getToken

Description:

Method	Return values	Description
getToken(String token, long tokenId)	Token	Return a token object.
The parameter tokenId is a valid token id.		

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMWorkflow;  
import com.openkm.bean.workflow.Token;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            long tkId = 8110; // Some valid token id  
            Token tk = OKMWorkflow.getInstance().getToken(null, tkId);  
            System.out.println(tk);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

addTokenComment

Description:

Method	Return values	Description
addTokenComment(String token, long tokenId, String message)	void	Add a token comment.
The parameter tokenId is a valid token id.		

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMWorkflow;  
import com.openkm.bean.workflow.Token;  
  
public class Test {
```

```

    public static void main(String[] args) {
        try {
            long tkId = 8110; // Some valid token id
            Token tk = OKMWorkflow.getInstance().addTokenComment(null, tkId, "here th
            System.out.println(tk);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

suspendToken

Description:

Method	Return values	Description
suspendToken(String token, long tokenId)	void	Suspend a token.
The parameter tokenId is a valid token id.		

Example:

```

package com.openkm;

import com.openkm.api.OKMWorkflow;
import com.openkm.bean.workflow.Token;

public class Test {
    public static void main(String[] args) {
        try {
            long tkId = 8110; // Some valid token id
            Token tk = OKMWorkflow.getInstance().suspendToken(null, tkId);
            System.out.println(tk);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

resumeToken

Description:

Method	Return values	Description
resumeToken(String token, long tokenId)	void	Resume a token.
The parameter tokenId is a valid token id.		

Example:

```
package com.openkm;

import com.openkm.api.OKMWorkflow;
import com.openkm.bean.workflow.Token;

public class Test {
    public static void main(String[] args) {
        try {
            long tkId = 8110; // Some valid token id
            Token tk = OKMWorkflow.getInstance().resumeToken(null, tkId);
            System.out.println(tk);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

sendTokenSignal

Description:

Method	Return values	Description
<i>sendTokenSignal(String token, long tokenId, String transitionName)</i>	<i>Token</i>	<i>Send a signal.</i>
<p><i>The parameter tokenId is a valid token id.</i></p> <p><i>The parameter transitionName is a transaction name.</i></p>		

Example:

```
package com.openkm;

import com.openkm.api.OKMWorkflow;
import com.openkm.bean.workflow.Token;

public class Test {
    public static void main(String[] args) {
        try {
            long tkId = 8110; // Some valid token id
            Token tk = OKMWorkflow.getInstance().sendTokenSignal(null, tkId, "end");
            System.out.println(tk);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

setTokenNode

Description:

Method	Return values	Description
<i>setTokenNode(String token, long tokenId, String nodeName)</i>	<i>void</i>	<i>Set token node.</i>
<i>The parameter tokenId is a valid token id.</i>		
<i>The parameter nodeName is a node name.</i>		

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMWorkflow;  
import com.openkm.bean.workflow.Token;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            long tkId = 8110; // Some valid token id  
            OKMWorkflow.getInstance().setTokenNode(null, tkId, "formA");  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

endToken

Description:

Method	Return values	Description
<i>endToken(String token, long tokenId)</i>	<i>Token</i>	<i>End token.</i>
<i>The parameter tokenId is a valid token id.</i>		

Example:

```
package com.openkm;  
  
import com.openkm.api.OKMWorkflow;  
import com.openkm.bean.workflow.Token;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            long tkId = 8110; // Some valid token id  
            OKMWorkflow.getInstance().setTokenNode(null, tkId);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
}  
  }  
}
```


Commons classes

OpenKM commons classes:

OpenKM common class	Description
ArchiveUtils	Utility methods to analyze a archive. For example create zip or create jar.
DbSessionManager	Utility methods used to manage user session.
ExecutionUtils	Utilities of execution methods. For example run script or run jar.
FileLogger	Utility methods to analyze a logger. For example show error, info or warning
FileUtils	Utility methods to analyze a file. For example get name or get parent or get extension.
FormatUtil	Utility methods used by formatters. For example format size or format parse size or format time.
ISO8601	Utility class what provides bidirectional format transformation between java time objects and ISO8601 format used for metadata date fields.
LegacyDAO	Manage the execution of queries in bdd. For example execution of queries in SQL or HQL.
MailUtils	Utility methods to analyze a mail. For example send message or send document.
PathUtils	Utility methods to analyze a route. For example, get the name or get the parent.
PDFUtils	Utility methods to convert pdf files. For example stamp image or stamp text.
PrincipalUtils	Utility methods for retrieving information from the security context. For example, get the actual userId from the security context session or get the actual tenant of the logged user.
StackTraceUtils	Utility methods for the stack.

<i>TemplateUtils</i>	<i>Utility methods to analyze a template. For example template existence or replace.</i>
<i>WorkflowUtils</i>	<i>Utility methods to handle workflows.</i>

ArchiveUtils

Utility methods for archive handling. For example, to create a ZIP or JAR.

Methods

createZip

Description:

Method	Return values	Description
createZip(File path, OutputStream os)	void	Create ZIP archive from file.
path: The file to compress.		
os: The output stream of the file.		

Example:

```
package com.openkm;  
  
import java.io.ByteArrayOutputStream;  
import java.io.File;  
  
import com.openkm.util.ArchiveUtils;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            File file = new File("/home/openkm/test.pdf");  
            ByteArrayOutputStream baos = new ByteArrayOutputStream();  
            ArchiveUtils.createZip(file, baos);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}  
}/code>
```

createJar

Description:

Method	Return values	Description
createJar(File path, String root, OutputStream os)	void	Recursively create JAR archive from directory.
path: The file to compress.		

root: A text with destiny of the file.

os: The output stream of the file.

Example:

```
package com.openkm;

import java.io.File;
import java.io.FileOutputStream;
import java.io.StringWriter;

import com.openkm.util.ArchiveUtils;
import com.openkm.util.FileUtils;
import com.openkm.util.PathUtils;
import com.openkm.util.impexp.RepositoryExporter;
import com.openkm.util.impexp.TextInfoDecorator;

public class Test {
    public static void main(String[] args) {
        try {
            StringWriter out = new StringWriter();

            final File tmpFile = File.createTempFile("/home/openkm/test", ".jar");
            FileOutputStream os = new FileOutputStream(tmpFile);

            File tmp = FileUtils.createTempDir();
            // Export files
            RepositoryExporter.exportDocuments(null, "/home/openkm", tmp, false, false, out,
                new TextInfoDecorator("/home/openkm"));

            // Jar files
            ArchiveUtils.createJar(tmp, PathUtils.getName("/home/openkm"), os);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

DbSessionManager

Utility methods used to manage user session. For example, to get the token of the super user system.

Methods

getSystemToken

Description:

Method	Return values	Description
getSystemToken()	String	Return the token of the super user system.

Example:

```
package com.openkm;  
  
import java.io.File;  
import com.openkm.module.db.stuff.DbSessionManager;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            System.out.println(DbSessionManager.getInstance().getInstance());  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

ExecutionUtils

Execution related methods. For example, to run a script or JAR.

Methods

runScript

Description:

Method	Return values	Description
runScript(File script)	Object[]	Execute script from file.
script: The name of the file to run.		

Example:

```
package com.openkm;  
  
import java.io.File;  
  
import com.openkm.core.Config;  
import com.openkm.util.ExecutionUtils;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            File script = new File("/home/openkm/test.bsh");  
            ExecutionUtils.runScript(script);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

runJar

Description:

Method	Return values	Description
runJar(File jar)	Object	Execute jar from file.
jar: The name of the file to run.		

Example:

```
package com.openkm;  
  
import java.io.File;  
  
import com.openkm.util.ExecutionUtils;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            File jar = new File("/home/openkm/test.jar");  
            ExecutionUtils.getInstance().runJar(jar);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

runCmd

Description:

Method	Return values	Description
runCmd(String cmd)	CmdExecutionResult	Execute command line.
cmd : The command line to execute.		

Example:

```
package com.openkm;  
  
import com.openkm.util.ExecutionUtils;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            ExecutionUtils.runCmd("mkdir test");  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

FileLogger

Utility methods to generate a log. For example, to show error, info or warning.

Methods

info

Description:

Method	Return values	Description
info(String baseName, String message, Object... params)	void	Static file logger with info.
baseName: The name of file to write.		
message: A text with message for info.		

Example:

```
package com.openkm;  
  
import com.openkm.util.FileLogger;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            FileLogger.info("test.log", "This is info");  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

warn

Description:

Method	Return values	Description
warn(String baseName, String message, Object... params)	void	Static file logger with warn.
baseName: The name of file to write.		
message: A text with message for warning.		

Example:

```
package com.openkm;  
  
import com.openkm.util.FileLogger;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            FileLogger.warn("test.log", "This is warning");  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

error

Description:

Method	Return values	Description
error(String baseName, String message, Object... params)	void	Static file logger with error.
baseName: The name of file to write.		
message: A text with message for error.		

Example:

```
package com.openkm;  
  
import com.openkm.util.FileLogger;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            FileLogger.error("test.log", "This is error");  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

FileUtils

File related utility methods. For example, to get the filename or extension.

Methods

getFileName

Description:

Method	Return values	Description
getFileName(String file)	String	Returns the name of the file without the extension.
file: The full name of the file.		

Example:

```
package com.openkm;  
  
import java.io.File;  
  
import com.openkm.util.FileUtils;  
import com.openkm.util.PathUtils;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            File file = new File("/home/openkm/test.png");  
            String docName = PathUtils.getName(file.getPath());  
            String baseName = FileUtils.getFileName(docName);  
            System.out.println(baseName);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

getFileExtension

Description:

Method	Return values	Description
getFileExtension(String file)	String	Returns the filename extension.
file: The full name of the file.		

Example:

```
package com.openkm;  
  
import java.io.File;  
  
import com.openkm.util.FileUtils;  
import com.openkm.util.PathUtils;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            File file = new File("/home/openkm/test.png");  
            String docName = PathUtils.getName(file.getPath());  
            String extension = FileUtils.getFileExtension(docName);  
            System.out.println(extension);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

FormatUtil

Utility methods used by formatters. For example, to format a file size or format time.

Methods

formatSize

Description:

Method	Return values	Description
formatSize(long bytes)	String	Format the document size for human readers.
bytes: The size of the file content.		

Example:

```
package com.openkm;  
  
import java.io.File;  
import com.openkm.util.FormatUtil;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            File file = new File("/home/openkm/test.png");  
            System.out.println(FormatUtil.formatSize(file.length()));  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

parseSize

Description:

Method	Return values	Description
parseSize(String text)	long	Parse human-readable sizes.
text: The size to parse.		

Example:

```
package com.openkm;
```

```
import com.openkm.util.FormatUtil;

public class Test {
    public static void main(String[] args) {
        try {
            System.out.println(FormatUtil.parseSize("1024.7"));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

formatDate

Description:

Method	Return values	Description
<i>formatDate(Calendar cal)</i>	<i>String</i>	<i>Format calendar date.</i>
<i>cal</i> : The date to format.		

Example:

```
package com.openkm;

import java.util.Calendar;
import com.openkm.util.FormatUtil;

public class Test {
    public static void main(String[] args) {
        try {
            System.out.println(FormatUtil.formatDate(Calendar.getInstance()));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

ISO8601

Utility class what provides bidirectional format transformation between java time objects and ISO8601 format used for metadata date fields.

Methods

parseBasic

Description:

Method	Return values	Description
parseBasic(String value)	String	Return date from an ISO8601 basic format string.
value: A ISO86101 basic format string value.		

Example:

```
package com.openkm;  
  
import com.openkm.util.ISO8601;  
import java.util.Calendar;  
import java.text.SimpleDateFormat;  
  
public class Test {  
    public static void main(String[] args) throws Exception {  
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");  
        Calendar cal = ISO8601.parseBasic("20180222000000");  
        System.out.println(sdf.format(cal.getTime()));  
    }  
}
```

formatBasic

Description:

Method	Return values	Description
formatBasic(Calendar value)	Calendar	Return ISO8601 basic format from date.
value: The date to format.		

Example:

```
package com.openkm;  
  
import com.openkm.util.ISO8601;  
import java.util.Calendar;  
  
public class Test {  
    public static void main(String[] args) throws Exception {  
        Calendar cal = Calendar.getInstance();  
        System.out.println(ISO8601.formatBasic(cal.getTime()));  
    }  
}
```

LegacyDAO

Manage the execution of queries in database. For example, the execution of queries in SQL or HQL.

Methods

executeSQL

Description:

Method	Return values	Description
executeSQL(final String query)	List<List<String>>	Execute SQL query.
query: The text of the query to execut		

Example:

```
package com.openkm;

import java.util.List;

import com.openkm.bean.form.Option;
import com.openkm.dao.LegacyDAO;

public class Test {
    public static void main(String[] args) {
        try {
            String query = "select DMT_VIRTUAL_COLUMN, DMT_REAL_COLUMN from OKM_DB_METADATA_TY
for (List<String> row : LegacyDAO.executeSQL(query)) {
    Option option = new Option();
    option.setValue(row.get(0));
    option.setLabel(row.get(1));
    System.out.println("First value :" + row.get(0) + " Second value: " + row.get(1)
}
} catch (Exception e) {
    e.printStackTrace();
}
}
```

executeHQL

Description:

Method	Return values	Description
executeHQL(final String query)	List<object>	Execute HQL query.

query: *The text of the query to execut*

Example:

```
package com.openkm;  
  
import java.util.List;  
import com.openkm.dao.LegacyDAO;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            String query = "select dmt.virtualColumn, dmt.realColumn from DatabaseMetadataType  
            List<object> list = LegacyDAO.executeHQL(query);  
            for (Object obj : list) {  
                Object[] dt = (Object[]) obj;  
                System.out.println(  
                    "First value :" + String.valueOf(dt[0]) + " Second value: " + String.valueOf(dt  
                )  
            }  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

MailUtils

Utility methods to send a mail. For example to send message or send document.

Methods

sendMessage

Description:

Method	Return values	Description
sendMessage(String toAddress, String subject, String content)	void	Send mail without FROM addresses.
toAddress: The destination addresses for mail.		
subject: The mail subject.		
content: The mail body.		

Example:

```
package com.openkm;  
  
import com.openkm.util.MailUtils;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            MailUtils.sendMessage("noreply@openkm.com", "this is the subject", "This is the te.  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

sendMessage

Description:

Method	Return values	Description
sendMessage(String fromAddress, String toAddress, String subject, String content)	void	Send mail with FROM addresses.
fromAddress: The from addresses for mail.		

toAddress: The destination addresses for mail.

subject: The mail subject.

content: The mail body.

Example:

```
package com.openkm;

import com.openkm.util.MailUtils;

public class Test {
    public static void main(String[] args) {
        try {
            MailUtils.sendMessage("from_noreply@openkm.com", "to_noreply@openkm.com", "this is
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

sendDocuments

Description:

Method	Return values	Description
sendDocuments(String fromAddress, List<String> toAddresses, String subject, String text, List<String> docsPath)	MimeMessage	Send document to non-registered OpenKM users.
<p>fromAddress: The from addresses for mail.</p> <p>toAddress: The list of destination addresses for mail.</p> <p>subject: The mail subject.</p> <p>text: The mail body.</p> <p>docsPath: The path of the documents to be send.</p>		

Example:

```
package com.openkm;

import java.util.ArrayList;
import java.util.List;
```

```
import com.openkm.util.MailUtils;

public class Test {
    public static void main(String[] args) {
        try {
            List<String> docsPath = new ArrayList<>();
            List<String> listAddress = new ArrayList<>();
            listAddress.add("to_noreply@openkm.com");
            docsPath.add("/home/openkm/test.png");
            docsPath.add("/home/openkm/test2.png");
            docsPath.add("/home/openkm/test3.png");
            MailUtils.sendDocuments("from_noreply@openkm.com", listAddress, "this is the subject", docsPath);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

PathUtils

Utility methods to handle paths. For example, get the name or get the parent path.

Methods

getParent

Description:

Method	Return values	Description
getParent(String path)	String	Get parent node.
path: The path of the file.		

Example:

```
package com.openkm;  
  
import com.openkm.util.PathUtils;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            System.out.println(PathUtils.getParent("/okm:root/test.docx"));  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

getName

Description:

Method	Return values	Description
getName(String path)	String	Get name node.
path: The path of the file.		

Example:

```
package com.openkm;
```

```
import com.openkm.util.PathUtils;

public class Test {
    public static void main(String[] args) {
        try {
            System.out.println(PathUtils.getName("/okm:root/test.docx"));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

getContext

Description:

Method	Return values	Description
getContext(String path)	String	Get path context. For example "/okm:root/test.txt" -> "/okm:root".
path: The path of the file.		

Example:

```
package com.openkm;

import com.openkm.util.PathUtils;

public class Test {
    public static void main(String[] args) {
        try {
            System.out.println(PathUtils.getContext("/okm:root/test.txt"));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

PDFUtils

Utility methods to convert PDF files. For example, to stamp image or text.

Methods

merge

Description:

Method	Return values	Description
<code>merge(List<InputStream> inputs, OutputStream output)</code>	<code>void</code>	Merge several PDFs into a new one.
<p>inputs: List of inputs stream for merge.</p> <p>output: Output stream of the file.</p>		

Example:

```
package com.openkm;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;
import org.apache.commons.io.IOUtils;

import com.openkm.api.OKMDocument;
import com.openkm.bean.Document;
import com.openkm.util.PDFUtils;
import com.openkm.util.PathUtils;
import com.openkm.util.FileUtils;

public class Test {
    public static void main(String[] args) {
        FileOutputStream fos = null;
        FileInputStream fis = null;
        File tmp = null;
        try {
            List<InputStream> inputs = new ArrayList<InputStream>();
            tmp = FileUtils.createTempFile("pdf");
            fos = new FileOutputStream(tmp);

            inputs.add(OKMDocument.getInstance().getContent(null, "/home/openkm/test1.pdf", fa
            inputs.add(OKMDocument.getInstance().getContent(null, "/home/openkm/test2.pdf", fa

            // Merge document
            PDFUtils.merge(inputs, fos);

            // Create document in repository
            String fldPath = PathUtils.getParent("/home/openkm/test1.pdf"); // all documents a
```

```
String docPath = fldPath + "/docMerge";
fis = new FileInputStream(tmp);
OKMDocument.getInstance().createSimple(null, docPath, fis);
} catch (Exception e) {
    e.printStackTrace();
} finally {
    IOUtils.closeQuietly(fis);
    IOUtils.closeQuietly(fos);
    FileUtils.deleteQuietly(tmp);
}
}
}
```


PrincipalUtils

Utility methods for retrieving information from the security context. For example, get the actual `userId` from the security context session or get the actual tenant of the logged user.

Methods

`getUser`

Description:

Method	Return values	Description
<code>getUser()</code>	<code>String</code>	Get the <code>userId</code> of the the logged user.



The user must be logged before executing the methods.

Example:

```
package com.openkm;  
  
package com.openkm.test;  
  
import com.openkm.spring.PrincipalUtils;  
  
public class Test {  
    public static void main(String[] args) throws Exception {  
        System.out.println(PrincipalUtils.getUser());  
    }  
}
```

StackTraceUtils

Utility methods for the stack.

Methods

isCallingMe

Description:

Method	Return values	Description
isCallingMe(String className)	Boolean	Return true when a className is in the current stack

Example:

```
package com.openkm;  
  
package com.openkm.test;  
  
import com.openkm.spring.PrincipalUtils;  
  
public class Test {  
    public static void main(String[] args) throws Exception {  
        System.out.println(StackTraceUtils.isCallingMe("com.openkm.automation.action.U  
    }  
}
```

toString

Description:

Method	Return values	Description
toString(Throwable t)	String	Return exception stack as standard text.

Example:

```
package com.openkm;  
  
package com.openkm.test;  
  
import com.openkm.spring.PrincipalUtils;  
  
public class Test {  
    public static void main(String[] args) throws Exception {  
        try {  
            // Force exception  
            List keys = null;  
            for (String key : keys) {
```

```
    }  
        } catch (Exception e) {  
            System.out.println(StackTraceUtilsUtils.toString(e));  
        }  
    }  
}
```

TemplateUtils

Utility methods to handle templates. For example, replace a text in a template.

Methods

templateExists

Description:

Method	Return values	Description
templateExists(String name)	boolean	Check for template existence.
name: The name of the template.		

Example:

```
package com.openkm;  
  
import com.openkm.util.TemplateUtils;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            System.out.println("Exist template: " + TemplateUtils.templateExists("OpenKM - NOT  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

replace

Description:

Method	Return values	Description
replace(String name, String template, Map<String, Object> model)	String	Quick replace utility function.
name: The name of the template.		
template: A text with the template definition.		
model: A map use to replace elements in the template.		

Example:

```
package com.openkm;  
  
import java.io.File;  
import java.util.HashMap;  
  
import com.openkm.core.Config;  
import com.openkm.util.FileUtils;  
import com.openkm.util.TemplateUtils;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            HashMap<String, Object> hm = new HashMap<String, Object>();  
            File tmpFileIn = FileUtils.createTempFileFromMime("text/plain");  
            hm.put("fileIn", tmpFileIn.getPath());  
            String cmd = TemplateUtils.replace("SYSTEM_XLS2CSV", Config.SYSTEM_CATDOC_XLS2CSV,  
            System.out.println("cmd template: " + cmd);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

WorkflowUtils

Utility methods to handle workflows.

Methods

findProcessInstancesByNode

Description:

Method	Return values	Description
<i>findProcessInstancesByNode(String uuid)</i>	<i>List<ProcessInstance></i>	Retrieves a list of all process instances of a node
<i>uuid</i> string type is the uuid of the document, folder, mail or record.		

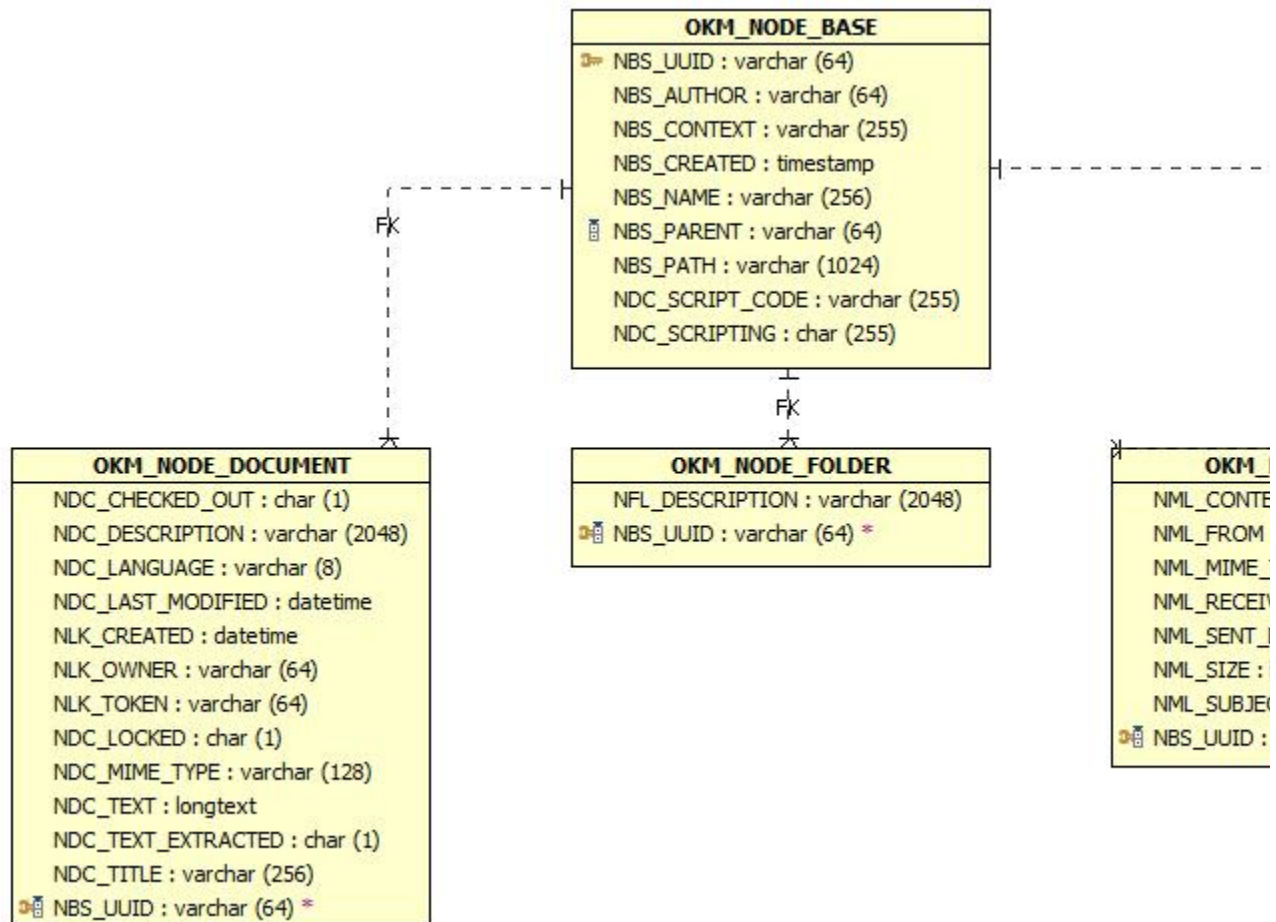
Example:

```
package com.openkm;  
  
import com.openkm.util.WorkflowUtils;  
import com.openkm.bean.workflow.ProcessInstance;  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            String uuid = "some node uuid";  
            for (ProcessInstance processInstance : WorkflowUtils.findProcessInstances.  
                System.out.println(processInstance);  
            }  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Database description

Node types

These tables store the information about a node in OpenKM.



The table named **OKM_NODE_BASE** store the common fields (name, author, creation date, among others) for all the node types (Document, Folder or Mail).

The field named **NBS_UUID** is the **unique node ID**. **NBS_UUID FIELD** is the **foreign key between parent table** (**OKM_NODE_BASE**) **and childs** (**OKM_NODE_DOCUMENT**, **OKM_NODE_FOLDER**, **OKM_NODE_MAIL**).

There are other useful tables:

- **OKM_NODE_DOCUMENT**: Store information about documents.
- **OKM_NODE_FOLDER**: Store information about folders.
- **OKM_NODE_MAIL**: Store information about mails.

SQL example queries

Query used to retrieve the information about every document:

```
select * from okm_node_base nb
inner join okm_node_document nd on nd.nbs_uuid = nb.nbs_uuid
```

Query to get folders what the name starts with 'test':

```
select * from okm_node_base nb
inner join okm_node_folder nr on nr.nbs_uuid = nb.nbs_uuid
where nb.nbs_name like 'test%'
```

To find a Document by uuid could use this query:

```
select * from okm_node_base nb
inner join okm_node_document nd on nd.nbs_uuid = nb.nbs_uuid
where nb.nbs_uuid = 'fc35ce80-57f2-4026-ae77-7a11bca042ed'
```

Mark documents to being indexed again

The column named `NDC_TEXT_EXTRACTED` in the table `OKM_NODE_DOCUMENT` it takes control of what documents has been processed into the Text extraction queue and what are still pending to be procesed (value 'T' means processed and 'F' means pending). If you want to reindex the whole repository again or some files you might execute a query like this one:

```
update OKM_NODE_DOCUMENT set NDC_TEXT_EXTRACTED='F'
```

Node version

The table named `OKM_NODE_DOCUMENT_VERSION` store the information of all versions of a document.

OKM_NODE_DOCUMENT	OKM_NODE_DOCUMENT_VERSION
NDC_CHECKED_OUT : char (1)	NDV_UUID : varchar (64)
NDC_DESCRIPTION : varchar (2048)	NDV_AUTHOR : varchar (64)
NDC_LANGUAGE : varchar (8)	NDV_CHECKSUM : varchar (32)
NDC_LAST_MODIFIED : datetime	NDV_COMMENT : varchar (2048)
NLK_CREATED : datetime	NDV_CONTENT : blob
NLK_OWNER : varchar (64)	NDV_CREATED : timestamp
NLK_TOKEN : varchar (64)	NDV_CURRENT : char (255)
NDC_LOCKED : char (1)	NDV_MIME_TYPE : varchar (128)
NDC_MIME_TYPE : varchar (128)	NDV_NAME : varchar (64)
NDC_TEXT : longtext	NDV_PARENT : varchar (64)
NDC_TEXT_EXTRACTED : char (1)	NDV_PREVIOUS : varchar (64)
NDC_TITLE : varchar (256)	NDV_SIZE : bigint (20)
NBS_UUID : varchar (64)	

The field named `NDV_UUID` is the **unique document version ID**.

The value of the field named **NDV_PARENT** is the **unique document ID** (the same value as **NBS_UUID**).

SQL example queries

Query TO get all the versions of a document named 'test.pdf':

```
select * from okm_node_document nd
inner join okm_node_document_version nv on nv.ndv_parent = nd.nbs_uuid
inner join okm_node_base nb on nd.nbs_uuid = nb.nbs_uuid
where nb.nbs_name = 'test.pdf'
```

This query is useful to know where all the versions of a file named 'test.pdf' are stored into the datastore:



Only applicable to the default datastore configuration (case all the versions of the documents are stored in the file system).

Take a look at [Datastore](#) for more information.

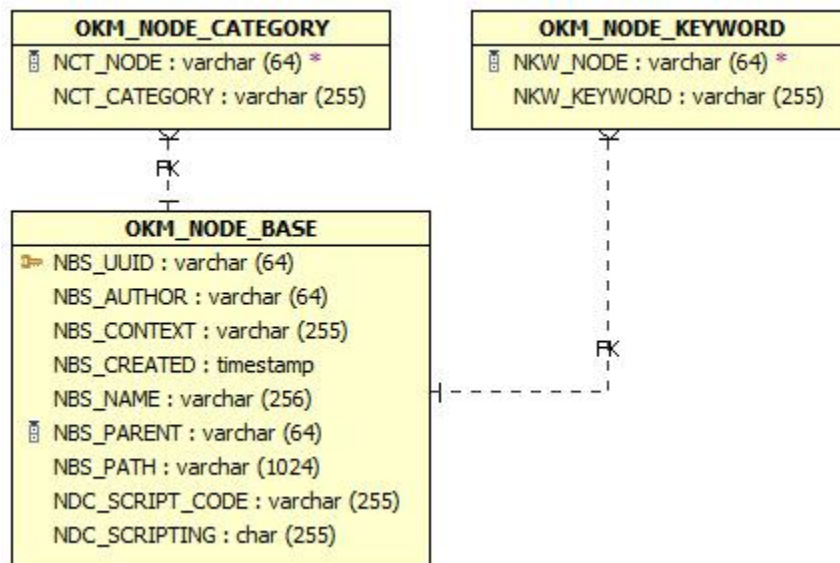
```
select ndv_uuid from okm_node_document nd
inner join okm_node_document_version nv on nv.ndv_parent = nd.nbs_uuid
inner join okm_node_base nb on nd.nbs_uuid = nb.nbs_uuid
where nb.nbs_name = 'test.pdf'
```



For example 'f0d7f0a1-b337-4917-b592-56e163b81cde'. This means your file is located in the physical drive in the path (based in first eight characters of this id):

/home/user/tomcat_home/repository/datastore/f0/d7/f0/a1/f0d7f0a1-b337-4917-b592-56e163b81cde

Node categories and keywords



The value of the field named **NCT_NODE** is the **unique document ID** (the same value as **NBS_UUID**).

The value of the field named **NKW_NODE** is the **unique document ID** (the same value as **NBS_UUID**).

SQL example queries

Query to extract all keywords associated with a node named 'test.pdf':



```
select * from okm_node_base nb
inner join okm_node_keyword nk on nk.nkw_node = nb.nbs_uuid
where nb.nbs_name = 'test.pdf'
```


Query to get all the categories for a node named 'test.pdf':

```
select * from okm_node_base nb
inner join okm_node_category nc on nc.nct_node = nb.nbs_uuid
where nb.nbs_name = 'test'
```

Node properties

In the table named **OKM_NODE_PROPERTY** are stored the metadata of a node (Document, Folder, Mail, Record).

OKM_NODE_PROPERTY	
	NPG_ID : bigint (20) I
	NPG_GROUP : varchar (64)
	NPG_NAME : varchar (64)
	NPG_VALUE : varchar (512)
	NPG_NODE : varchar (64)

OKM_NODE_DOCUMENT	
	NDC_CHECKED_OUT : char (1)
	NDC_DESCRIPTION : varchar (2048)
	NDC_LANGUAGE : varchar (8)
	NDC_LAST_MODIFIED : datetime
	NLK_CREATED : datetime
	NLK_OWNER : varchar (64)
	NLK_TOKEN : varchar (64)
	NDC_LOCKED : char (1)
	NDC_MIME_TYPE : varchar (128)
	NDC_TEXT : longtext
	NDC_TEXT_EXTRACTED : char (1)
	NDC_TITLE : varchar (256)
	NBS_UUID : varchar (64)

The value of the field named **NPG_NODE** value is the **unique document ID** (the same value as **NBS_UUID**).

The field named **NPG_GROUP** is the **metadata group name** (for example okg:consulting)

The field named **NPG_NAME** is the **metadata field name** (for example okp:consulting.comment)

SQL example queries

Query to get all the metadata associated with a document named 'test.pdf':

```
select nd.nbs_uuid, nbs_name, npg_group, npg_name, npg_value
from okm_node_document nd
inner join okm_node_base nb on nd.nbs_uuid = nb.nbs_uuid
inner join okm_node_property np on np.npg_node = nd.nbs_uuid
```

```
where nb.nbs_name = 'test.pdf'
```