

Spécifications fonctionnelles du client d'administration du serveur

Spécifications fonctionnelles du client d'administration du serveur

WebDAV ESUP

**

Auteur : Thomas Bellebois - Université de Rennes 1 (<http://>)

- [Spécifications fonctionnelles du client d'administration du serveur](#)
- [Versions](#)
- [Introduction](#)
- [Spécifications fonctionnelles - description générale](#)
- [Spécifications fonctionnelles - description détaillée](#)
 - [Interface utilisateur](#)
 - [Lancement du programme](#)
 - [Modes interactif et fichier - commandes - syntaxe et](#)
 - [Mode interactif - aperçu](#)
 - [Mode interactif - création d'espaces avec ACL et](#)
 - [Mode interactif - gestion des quotas](#)
 - [Mode fichier](#)
 - [Mode graphique](#)
- [Spécifications techniques](#)
 - [Spécifications de l'outil "spaceCreation"](#)
 - [Introduction](#)
 - [Syntaxe](#)
 - [Fonctionnalités](#)
 - [Fichier template](#)
 - [Structure du fichier](#)
 - [Exemples](#)
 - [Création d'un répertoire classique sans droits ni](#)
 - [Création d'un répertoire classique avec droits et](#)
 - [Création d'une série de homedirs avec hashage](#)

Versions

- 12.10.2005 - Thomas Bellebois (Rennes1) - Première version - v1.0
- 25.10.2005 - Thomas Bellebois (Rennes1) - Ajout des modes fichier et graphique - ajout des spécifications techniques - mise à jour de l'existant - v2.0
- 08.11.2005 - Thomas Bellebois (Rennes1) - Suppression des références restantes aux outils homedirCreation et inJacSpacesCreation - Changement du nom de l'outil folderCreation en spaceCreation- Modification du fichier de templates de l'outil spaceCreation - corrections mineures - v2.1
- 08.11.2005 - Thomas Bellebois (Rennes1) - Correction d'un bug de spécification concernant la métadonnée protected - v2.1.1

Introduction

Le client d'administration du serveur WebDAV intégré à partir de la version 4.0 du serveur regroupe d'anciens outils d'administration et en ajoute de nouveaux. Il permet notamment la navigation dans les arborescences, la manipulation de répertoires, la manipulation de quotas et la création d'espaces avec ACL et métadonnées. Ce client est utilisable en mode interactif (interpréteur de commande), en mode ligne de commande (en fournissant un fichier de commandes en entrée) ou grâce à une IHM Java.

Dans la version 4.0 seule les modes interactif et ligne de commande sont disponibles.

Ce document n'est pas un manuel d'utilisation et ne décrit donc que succinctement le programme.

Spécifications fonctionnelles - description générale

Les fonctionnalités du client sont : * Création d'espaces avec ACL et métadonnées (permettant donc de créer des homedirs protégés, espaces partagés, espaces InJac)

- Gestion de quotas (interrogation, positionnement et suppression des quotas sur une arborescence)
- Navigation dans les arborescences

Spécifications fonctionnelles - description détaillée

Interface utilisateur

Le mode ligne de commandes est soit interactif, soit récupère toutes les commandes à exécuter depuis un fichier de commandes.

Lancement du programme

Mode graphique : ESUP-WDServerManager gr

Mode interactif : ESUP-WDServerManager cl

Mode ligne de commande : ESUP-WDServerManager cl pathToCommandFile

Modes interactif et fichier - commandes - syntaxe et

fonctionnalités

Les paramètres entre parenthèses sont optionnels. Les commandes marquées par (com.file) sont utilisables dans le mode ligne de commande (dans le fichier d'entrée) et dans ce cas les paramètres optionnels doivent être renseignés.

Aperçu des commandes, de leur syntaxe et fonctionnalités :

```

->menu Display the menu.
->getConfig Display the application parameters stored in the properties/configuration.xml file.
->setConfig (com.file) syntax : setConfig ([property] [value]) Modify the application parameters stored in the
properties/configuration.xml. Note that these parameters can also be modified directly in the file (and restart
the application).
->connect (com.file) syntax : connect Connect to the WebDAV server and associated quota Web service. This
command MUST BE called first.
->treeInfo (com.file) syntax : treeinfo ([depth]) Displays the tree information from the current directory.
Displays children up to the given depth. >> resource consuming process with a large depth <<
->getQUB (com.file) syntax : getQUB ([path]) Get the quota-used-bytes metadata (in bytes) of the current
directory.
->getQAB (com.file) syntax : getQAB ([path]) Get the quota-available-bytes metadata (in bytes) of the current
directory.
->getVR (com.file) syntax : getVR ([path]) Get the virtual-root metadata of the current directory.
->patchQUB (com.file) syntax : patchQUB ([path] [value]) Patch the quota-used-bytes metadata of the current
directory (in bytes).
->patchQAB (com.file) syntax : patchQAB ([path] [value]) Patch the quota-available-bytes metadata of the
current directory (in bytes).
->setVR (com.file) syntax : setVR ([path] [value]) Set the virtual-root metadata of the current directory (true
or false).
->setQuota (com.file) syntax : setQuota ([path] [depth] [quota] [calcQUB]) Set resource quota. The depth
specifies where quota has to be set. Example : currentDirectory=/files depth=2 Quota will be set to every sub-
sub-directory of /files
->removeQuota (com.file) syntax : removeQuota ([path] [processChildren]) Remove resource quota. You can remove
children quota or not.
->calcQUB Calculate resource quota-used-bytes metadata. >> resource consuming process <<
->ls Display resource children.
->cd syntax : cd directoryName (relative or absolute path) Change directory. ex : cd /files/test, cd test, cd
test/myDir
->mkdir (com.file) syntax : mkdir [directoryName] Create a new directory.
->rmdir (com.file) syntax : rmdir (-f) [directoryName] Delete a directory -f : force deletion.
->createSpace (com.file) syntax : folderCreation -path=[pathToCreateTheFolder]
| -name=[folderName]
or |_-ldap=[ldapFilter] -ldapAttribute=
[uniqueLdapAttributeToRetrieve]
(-regexp=[regexp] -regexpSeparator=[regexpSeparator])
(-templateFile=[pathToTemplateFile])
Create a space with ACL and metadata.
->help Display the help. syntax : help [commandName] Parameters in brackets are optionnals The tag (com.file)
means that the command can be used in the input command file >> [commandName] is case sensitive <<
->quit (com.file) Quit.

```

Mode interactif - aperçu

!screenshotInteractif.jpg!h2. Mode interactif - configuration et connexion

Toute la configuration se trouve dans un fichier properties/configuration.xml et est modifiable soit en modifiant directement ce fichier ou via des commandes de l'interface. Notez que tous les messages de l'interface se trouvent dans des fichiers properties et sont donc personnalisables. Une commande connect permet de se connecter au serveur WebDAV et à son service web de gestion des quotas.

!screenshotConfiguration.jpg!h2. Mode interactif - navigation

Des fonctions de base permettent de naviguer dans une arborescence ainsi que de manipuler des répertoires.

!screenshotNavigation.jpg!La commande cd accepte des chemins absolus et relatifs ainsi que le ..

Mode interactif - création d'espaces avec ACL et

métadonnées

Cette fonctionnalité est expliquée en détails dans la partie 5.1 de ce document.

Mode interactif - gestion des quotas

Les commandes treeInfo [depth](#), getQUB, getQAB et getVR permettent d'afficher les métadonnées de quotas d'un répertoire ou d'une arborescence.

!screenshotTreeInfo.jpg!Les commandes patchQAB, patchQUB et setVR permettent de modifier les métadonnées de quotas d'un répertoire.

La commande setQuota permet de positionner le quota d'un ou d'une série de répertoires d'une profondeur donnée et calcul automatique possible de la place déjà utilisée par le répertoire.

!screenshotSetQuota.jpg!La commande removeQuota permet de supprimer le quota d'un répertoire (toutes les métadonnées de quota) avec traitement récursif sur les fils ou non.

Mode fichier

Les commandes du mode fichier sont exactement les même que celles du mode interactif. Les commandes du mode interactif nécessitant une intervention utilisateur possèdent aussi une syntaxe automatique.

exemple : la commande setQuota (cf. partie "Mode interactif - gestion des quotas") interactive possède aussi la syntaxe setQuota **path depth quota calcQU**

Les commandes sont entrées sur chaque ligne dans le fichier. Les caractère # et // permettent de commenter une ligne. Les caractères /* et */ permettent de commenter un bloc.

exemple d'un fichier de commandes :

```
#
# ESUP WebDAV server manager command file
#
# you can add comment like this
//or like this
/*
    or like this
*/
# --> connecting to the server
connect
# --> deleting the test tree if it exists
rmdir -f /files/quotas
# --> trying an impossible action (for test)
mkdir /files/repQuiExistePas/newRep
# --> building a new test tree
mkdir /files/quotas
mkdir /files/quotas/goodGuys
mkdir /files/quotas/goodGuys/jedi
mkdir /files/quotas/goodGuys/jedi/yoda
mkdir /files/quotas/goodGuys/jedi/quiGonJin
mkdir /files/quotas/badGuys
mkdir /files/quotas/badGuys/sith
mkdir /files/quotas/badGuys/sith/darkVador
mkdir /files/quotas/badGuys/sith/darkMaul
mkdir /files/quotas/badGuys/sith/darkMaul/babyDarkMaul
mkdir /files/quotas/badGuys/sith/darkMaul/doggyDarkMaul
# --> setting up quotas
setquota /files/quotas 0 7000 false
//for goodGuys badGuys
setquota /files/quotas 1 5000 false
//for yoda quiGonJin darkVador darkMaul
setquota /files/quotas 3 2000 false
setquota /files/quotas/badGuys/sith/darkMaul/babyDarkMaul 0 2000 false
//darkMaul is a VR
setvr /files/quotas/badGuys/sith/darkMaul true
# --> seeing the result
cd /files
treeinfo 5
# --> leaving the program
quit
```

Résultat du fichier de commandes :

```
/>Quota manager connected to http://localhost:8080/slide/services/QuotaWS WebDAV manager connected to http://localhost:8080/slide Homedir manager
connected to http://localhost:8080/slide Injac spaces manager connected to http://localhost:8080/slide />Directory /files/quotas deleted successfully
/>Directory creation failed : /files/repQuiExistePas/newRep ! />Directory /files/quotas created successfully />Directory /files/quotas/goodGuys created
successfully />Directory /files/quotas/goodGuys/jedi created successfully />Directory /files/quotas/goodGuys/jedi/yoda created successfully />Directory /files
/quotas/goodGuys/jedi/quiGonJin created successfully />Directory /files/quotas/badGuys created successfully />Directory /files/quotas/badGuys/sith
created successfully />Directory /files/quotas/badGuys/sith/darkVador created successfully />Directory /files/quotas/badGuys/sith/darkMaul created
successfully />Directory /files/quotas/badGuys/sith/darkMaul/babyDarkMaul created successfully />Directory /files/quotas/badGuys/sith/darkMaul
/doggyDarkMaul created successfully />Quota set successfully for /files/quotas /files/quotas>Quota set successfully for /files/quotas /files/quotas>Quota
set successfully for /files/quotas /files/quotas>Quota set successfully for /files/quotas/badGuys/sith/darkMaul/babyDarkMaul /files/quotas/badGuys/sith
/darkMaul/babyDarkMaul>Metadata patched succesfully for /files/quotas/badGuys/sith/darkMaul /files/quotas/badGuys/sith/darkMaul>/files>| / [ QAB=0
QUB=0 VR=null ] | testQuota | quotas [ QAB=7000 QUB=0 VR=null ] | goodGuys [ QAB=5000 QUB=0 VR=null ] | jedi | yoda [ QAB=2000 QUB=0 VR=null
] | quiGonJin [ QAB=2000 QUB=0 VR=null ] | badGuys [ QAB=5000 QUB=0 VR=null ] | sith | darkVador [ QAB=2000 QUB=0 VR=null ] | darkMaul [
QAB=2000 QUB=0 VR=>>>true<<< ] | babyDarkMaul [ QAB=2000 QUB=0 VR=null ] | doggyDarkMaul
```

Mode graphique

TODO

Spécifications techniques

Spécifications de l'outil "spaceCreation"

Introduction

Ce nouvel outil remplace les outils "homedirCreation", "injacSpacesCreation" et "shareSpacesCreation" (ce dernier n'ayant jamais été implémenté, juste spécifié).

Il s'agit d'un outil générique permettant de créer des collections (répertoires) sur le serveur et d'y positionner des droits et métadonnées.

Syntaxe

```
spaceCreation
  -path=[pathToCreateTheFolder]
  | -name=[folderName]
ou |_-ldap=[ldapFilter] -ldapAttribute=[uniqueLdapAttributeToRetrieve]
  (-regexp=[regexp] -regexpSeparator=[regexpSeparator])
  (-templateFile=[pathToTemplateFile])
```

Les paramètres entre parenthèses sont optionnels.

Fonctionnalités

-path

Spécifie le chemin du répertoire père du répertoire à créer. Ce père doit déjà exister.

-name=**folderName** ou -ldap=**ldapFilter** -ldapAttribute=**uniqueLdapAttributeToRetrieve**

Permet de spécifier ou récupérer le nom du ou des répertoires à créer.

+ Soit le nom est spécifié par le paramètre -name (création d'un seul répertoire).

+ Soit il(s) est(sont) récupéré(s) dynamiquement depuis un annuaire LDAP avec les paramètres -ldap et -ldapAttribute. Le paramètre -ldapAttribute spécifie l'attribut (unique!) LDAP avec lequel construire le nom du répertoire.

-regexp=**regexp** -regexpSeparator=**regexpSeparator**

Paramètre optionnel permettant de spécifier un chemin d'accès au répertoire (en plus du chemin spécifié par le paramètre -path). Par défaut le(s) répertoire(s) est(sont) créé(s) en rateau directement dans le répertoire spécifié par -path. Ce paramètre permet par exemple de réaliser un hashage pour créer un grand nombre de répertoire afin d'optimiser les performances de stockage. _Notez que tous les répertoires sont créés automatiquement s'ils n'existent pas déjà._

-templateFile=[pathToTemplateFile](#)

Permet de fournir un fichier de template (ACL et métadonnées) à appliquer au **×** nouveau **×** répertoire(s).

_Utilisation de l'attribut d'une recherche LDAP

:_

Le résultat d'une recherche LDAP (lors de l'utilisation des attributs -ldap et -ldapAttribute) peut être utilisé :

- dans le paramètre -regexp pour contruire l'expression régulière basée sur le résultat de la recherche
- dans le fichier de template

Fichier template

Ce fichier permet de fournir un jeu d'ACL et de métadonnées à appliquer aux répertoires à créer.

Il contient deux sections properties et permissions, l'une des deux sections peut être absente (pour n'appliquer que des ACL ou que des métadonnées), voir les deux (fichier vide - pour ne rien appliquer).

Structure du fichier

```
<properties>
  <property name=[propertyName] (namespace=[namespace]) value=[value] (type=[type]) (protected=true|false) />
  <property ... />
  ...
</properties>
<permissions>
  <permission subjectUri=[subjectUri] actionUri=[actionUri] negative=true|false />
  <permission ... />
  ...
</permissions>
```

Les paramètres entre parenthèses sont optionnels.

La balise <properties> permet de définir les métadonnées à appliquer aux répertoires :

- name : nom de la métadonnée
 - namespace : espace de nom de la métadonnée - par défaut DAV:
 - value : valeur de la métadonnée
 - type : type de la métadonnée (peut être vide) - par défaut vide
- La balise <permissions> permet de définir les droits à appliquer aux répertoires :

- subjectUri : principal auquel s'applique la permission
 - actionUri : type et URI de la permission au sens Slide
 - negative : si negative=true alors la permission est une interdiction (DENY), sinon c'est une autorisation (GRANT) - par défaut false
 - protected : spécifie si la métadonnée est protégée - par défaut false
- Valeurs possibles pour le champs actionUri de la balise permissions>permission

```
/actions/read
/actions/write
/actions/write-acl
/actions/read-acl
/actions/read-current-user-privilege-set
/actions/unlock
/actions/write-properties
/actions/write-content
/actions/bind
/actions/unbind
```

Valeurs possibles pour le champs subjectUri de la balise permissions>permission

```
all : tout le monde
authenticated : utilisateur authentifié
unauthenticated : utilisateur non authentifié
self : // role pas bien défini
owner : propriétaire de la ressource
/users/john : utilisateur john
/roles/admin : role admin
```

Exemples

Création d'un répertoire classique sans droits ni

métadonnées
Fichier template

```
<!-- Ce fichier ne sera pas passé en paramètre du programme -->
```

Appel du programme

```
spaceCreation -path="/files/documentsESUP/projetINJAC" -name="canalStockage"
```

Création d'un répertoire classique avec droits et

métadonnées
Fichier template

```
<properties>
  <property name="displayname" value="canalStockage-YC-Valenciennes" />
</properties>
<permissions>
  <permission subjectUri="/users/ycolmant" actionUri="all" negative="false" />
  <permission subjectUri="/roles/groupeINJAC" actionUri="/actions/read" negative="false" />
  <permission subjectUri="all" actionUri="all" negative="true" />
</permissions>
```

Appel du programme

```
spaceCreation -path="/files/documentsESUP/projetINJAC" -name="canalStockage" -templateFile="/usr/local/src/templates/canalStockage-YC-Valenciennes_TEMPLATE.xml"
```

Création d'une série de homedirs avec hashage

Fichier template

```
<properties>
  <property name="owner" namespace="DAV:" value="{0}" protected=true />
</properties>
<permissions>
  <permission subjectUri="owner" actionUri="all" negative="false" />
  <permission subjectUri="all" actionUri="all" negative="true" />
</permissions>
```

Appel du programme

```
spaceCreation -path="/slide/files" -name="canalStockage" -ldap="((&(type=pers)(department=123))(uid=55*))" -ldapAttribute="uid" -templateFile="/usr/local/src/templates/canalStockage-YC-Valenciennes_TEMPLATE.xml" -regexp="/slide/files/:(.)/slide/files/:(.)/slide/files/:(.)/slide/files/(:)*" -regExpSeparator=":"
```

Conséquence de cette ligne de commande : * les entrées uid correspondantes au filtre spécifié sont remontées du LDAP

- pour chaque uid le path (/slide/files dans notre exemple) est concaténé avec l'uid pour former une chaîne de type /slide/files/[uid](#) - dans notre exemple /slide/files/ycolmant
- la regexp est appliquée à la chaîne précédemment formée pour former le chemin complet final du répertoire à créer. Dans notre exemple ce chemin sera par exemple /slide/file/yc/ycolmant

Notez que la métadonnée owner est protégée.