

User guide - esup-webdav-srv

ESUP WebDAV server

User guide

Auteur : ([http://])

- [ESUP WebDAV server](#)
- [Introduction](#)
- [Important notes](#)
- [Pre-requisite](#)
- [Naming conventions](#)
- [User guide](#)
 - [Deploying and configuring the tool](#)
 - [Changing the configuration](#)
 - [Recommended method](#)
 - [Alternate method](#)
 - [Running the tool](#)
 - [Execution modes](#)
 - [Interactive mode](#)
 - [Input command file mode](#)
 - [Commands overview](#)
 - [Commands details](#)
 - [Basic commands](#)
 - [Folder creation/deletion](#)
 - [Quota commands](#)
 - [Space manager command](#)
 - [Introduction](#)
 - [Syntax](#)
 - [Template file](#)
 - [Examples](#)
 - [Creating a simple space without metadata nor](#)
 - [The ESUP storage channel \(V4.2 and higher\)](#)

Introduction

The purpose of this document is to explain how to use the ESUP WebDAV server. In particular it focuses on the administration tool : ESUPWDServerManager.

Important notes

Note that some actions done by the administration tool are resource consuming. These actions are identified by the [RC] tag in this document.

Do not forget that the DAV:quota-available-bytes metadata represents the space left on a folder and not the quota. The quota is the sum of the DAV:quota-used-bytes and DAV:quota-available-bytes metadata.

You are adviced of reading the [paper submitted to EUNIS 2006]media/ESUPWebDAV-eunis2006-extended-abstract.pdf] to understand how quota are implemented on the ESUP WebDAV server.



About the ESUP storage channel : the v4.2 storage channel requires to use the ESUP WebDAV server V3.5 or higher and to have specific permissions and metadata set on homedirs. Please read carefully the section dealing with this topic both in this documentation and in the ESUP storage channel user guide.

Pre-requisite

- an ESUP WebDAV server deployed and running

Naming conventions

- [package.home] : directory where the package is unzipped.
- [deploy.home] : directory where the server is deployed.

User guide

Deploying and configuring the tool

ESUPWDServerManager is automatically deployed and configured with the ESUP WebDAV server.
folders * source : Java source files

- properties : properties files, sample input file and space manager sample pattern file
- lib : libraries
- dist : package directory
- build : execution directory

Changing the configuration

Recommended method

Modify the [package.home]/esup-serveur-WebDav-3.5/build.properties files and execute the init ANT target. Note that this action can *also modify the server configuration*. These parameters are marked with the [S] flag in the following list.

build.properties file parameters involving ESUPWDServerManager :

- webdavServer.url
- slide.Idap.Url : [S]
- slide.Idap.Port : [S]
- slide.Idap.Scope : [S]
- slide.Idap.Principal : [S]
- slide.Idap.Credential : [S]
- slide.Idap.MaxResults : [S]
- slide.Idap.BaseDn : [S]
- slide.quotaListener.WebServiceUrl
- slide.rootUser : [S]
- axis.trustedFilter.password

Alternate method

Modify the [deploy.home]/properties/configuration.xml file. Note that if you execute the init ANT target ([package.home]/esup-serveur-WebDav-4.0beta directory), your modifications will be lost.

Running the tool

Execution modes

ESUPWDServerManager can be either runned in interactive mode (like an Unix shell) or providing an input command file.

It is planned to develop a graphic mode.



Edit the run.bat or run.sh scripts (to specify paths).



The run scripts copy (and overwrite) the [deploy.home]/ESUPWDServerManager-1.0/properties directory and the log4j.xml file in the execution directory ([deploy.home]/ESUPWDServerManager-1.0/build).

Interactive mode

Execute the run.sh or run.bat script in the [deploy.home]/esup-serveur-WebDav-3.5/ESUPWDServerManager-1.0 directory :

./run.sh

Input command file mode

Execute the run.sh or run.bat script in the [deploy.home]/esup-seurveur-WebDav-3.5/ESUPWDServerManager-1.0/build directory followed by the input file path:

./run.sh /data/ESUPStuff/commands.txt

You will find commands.txt sample command file in the [deploy.home]/esup-seurveur-WebDav-3.5/ESUPWDServerManager-1.0/properties directory :

```
#
# ESUP WebDAV server manager command file
#
# you can add comment like this
//or like this
/*
    or like this
*/
# --> modifying some parameters
setConfig connectUserName="tbellemb"
setConfig connectPassword="trusted"
# --> connecting to the server
connect
# --> deleting the test tree
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 true
//darkMaul is a VR
setVR "/files/quotas/badGuys/sith/darkMaul" true
# --> creating spaces with metadata
createSpace -path="/files/quotas" -ldap="(&(&(urltypeentree=pers)(departmentNumber=57SI*)) (uid=b*))" -
ldapAttribute="uid"
        -templateFile="properties/folderManager_pattern.xml" -regexp="^(.).*:^(..)*:^(.*)" -
regexpSeparator=":"
createSpace -path="/files/quotas" -name="myNewSpace"
# --> setting some properties
patchQAB "/files/quotas/badGuys/sith/darkMaul/babyDarkMaul" 2500
# --> getting some properties
getQAB "/files/quotas/badGuys/sith/darkMaul/babyDarkMaul"
# --> removing quota
removeQuota "/files/quotas" false
# --> displaying the result
cd /files
treeinfo 5
# --> leaving the program
quit
```

Commands overview

!commandsOverview.png!ESUPWDServerManager was made to work almost like a Unix shell, it means that you can navigate using

Unknown macro: {span}

ls

and

Unknown macro: {span}

cd

commands and create/delete directories with

Unknown macro: {span}

mkdir

and

Unknown macro: {span}

rmdir

commands.

Most of the commands have several syntax so that they can be use both with the interactive or input file mode.

The first command to run is

Unknown macro: {span}

connect

to connect to the ESUP WebDAV server.

The

Unknown macro: {span}

help full

command gives the list and syntax of every command.

Commands details

For a full command help type

Unknown macro: {span}

help

[commandName]

.

Optional arguments are in brackets. Optional arguments have to be fill in with the input command file mode. In the interactive mode, if you do not provide the optional arguments, you will be prompted.

Basic commands

- Unknown macro: {span}

connect

- Unknown macro: {span}

quit

- Unknown macro: {span}

ls

: list current directory content

- Unknown macro: {span}
- cd
: changes current directory (relative or full path)
- Unknown macro: {span}
- menu
: displays the menu
- Unknown macro: {span}
- help
: displays the help (help full - help [commandName])

Folder creation/deletion

- Unknown macro: {span}
- mkdir
: creates a directory (relative or full path)
- Unknown macro: {span}
- rmdir
: deletes a directory (relative or full path)

Quota commands

- Unknown macro: {span}
- treeinfo
[RC] : advanced ls command showing children and quota metadata of the current directory up to the given depth- syntax : treeinfo [depth]
!treeInfo.png!In the screenshot above the /files/quotas/samples/100k directory has a quota of 66500 + 33500 bytes = 100kb with 33500bytes used.
- Unknown macro: {span}
- getQUB
: displays the folder DAV:quota-used-bytes metadata - syntax getQUB ("fullPath")
- Unknown macro: {span}
- getQAB
: displays the folder DAV:quota-available-bytes metadata - syntax getQAB ("fullPath")
- Unknown macro: {span}
- getVR
: displays the folder ESUP:virtual-root metadata - syntax getVR ("fullPath")
- Unknown macro: {span}
- patchQUB
: changes the folder DAV:quota-used-bytes metadata - syntax patchQUB ("fullPath value") - You should not have to use this command given that when you create a new folder the DAV:quota-used-bytes metadata can be automatically calculated.
- Unknown macro: {span}
- patchQAB
: changes the folder DAV:quota-available-bytes metadata - syntax patchQAB ("fullPath" value)
- Unknown macro: {span}
- setVR
[RC] : changes the folder ESUP:virtual-root metadata - automatically changes quota metadata of parent directories up to the root (or first virtual root) - syntax setVR ("fullPath" true/false)
- Unknown macro: {span}
- setQuota
[RC] : changes the folder quota or quota of its children at a given depth. Can calculate the DAV:quota-used-bytes metadata if directories contain data. syntax setQuota ("fullPath" depth quota calculateQUB) - depth=0 means that you set the quota of the current directory - calculateQUB=true/false!setQuota.png!

Space manager command

Introduction

This new tool replaces the former "HomedirCreation" and "InjacSpacesCreation" tools.

This is a generic tool to create folders and set permissions and metadata.

Syntax

```
createSpace
    -path="spacePath"
    | -name="spaceName"
or |_-ldap="ldapFilter" -ldapAttribute="uniqueLdapAttributeToRetrieve"
    (-regex="regex" -regexSeparator="regexSeparator" (-regexPattern="regexPattern"))
    (-templateFile="pathToTemplateFile")
    (-forceSetProperties="true|false")
```

-path

Specifies the space path. This path must exist. Do not specify the context !

-name="folderName" or -ldap="ldapFilter" -ldapAttribute="uniqueLdapAttributeToRetrieve"

Specifies or retrieves name(s) of directory(ies) to create.

+ Either the name is specified with the -name attribute (creation of only one space).

+ Or they are retrieved from an LDAP with the -ldap and -ldapAttribute parameters. The -ldapAttribute parameter specifies the unique LDAP attribute used for space names.

-regex="regex" -regexSeparator="regexSeparator"

Optional parameters to transform the full space paths.

With a regex you can specify a hash method to create homedirs and then improve the server performance. All of the sub-directories are automatically created.

-regexPattern="regexPattern"

Apply an additional transformation to "spaceName" or "uniqueLdapAttributeToRetrieve" before the regex mechanism.

Example: if uniqueLdapAttributeToRetrieve="uid" and for a user this uid value is "bourges" and regexPattern="

Unknown macro: {0}

/foo" then regex mechanism will work on "bourges/foo" string and not just "bourges"

-templateFile="pathToTemplateFile"

Specifies a template file to set permissions and metadata on spaces. Results retrieved from the LDAP can be used in the template file with the expression.

-forceSetProperties="true|false" (default=false)

Specifies if properties have to be written even if space(s) already exist. You can use this parameter to (re)initialize permissions and add metadata to existing space(s).

Template file

Contains two sections : permissions and properties. Remove one of them to put only permissions or metadata.

This file has to be compliant with the [deploy.home]/ESUPWDSERVERMANAGER1.0/properties/folderManager_pattern.dtd file :

```

<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT folderPattern (properties, permissions)>
<!ELEMENT properties (property)+>
<!ELEMENT permissions (permission)+>
<!ELEMENT property EMPTY>
<!ELEMENT permission EMPTY>
<ATTLIST property
  name CDATA #REQUIRED
  namespace CDATA #IMPLIED
  value CDATA #REQUIRED
  type CDATA #IMPLIED
>
<ATTLIST permission
  subjectUri CDATA #REQUIRED
  actionUri (read | write | read-acl | write-act | all) #REQUIRED
  negative (true | false) #REQUIRED
>

```

```

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

```

The <properties> tag defines metadata :

- name : property name - use only not protected property names except "owner"
- namespace : property namespace - by default DAV:
- value : property value
- type : property type - by default null

The <permissions> tag defines ACL's :

- subjectUri : ACL principal
- actionUri : action granted or denied
- negative : if true then the action is denied - by default false

Possible values for the actionUri tag :

```

read
write
write-acl
read-acl
all

```

Possible values for the subjectUri tag :

```

all : all of the users
authenticated : authenticated user
unauthenticated : unauthenticated user
self : // do not use this subject
owner : user owner
/users/john : a user
/roles/admin : a group

```

Examples

permissions
Template file

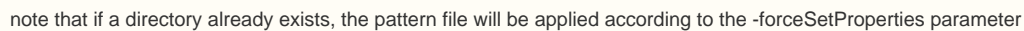
```
createSpace -path="/files/ESUP_documents" -name="WebDAVServer_documents" h6. Creating a simple space with metadata and permissions
Template file
```

```
createSpace -path="/files/ESUP_documents" -name="WebDAVServer_documents" -templateFile="/usr/local/src/templates/WebDAV_server-
Rennes1_TEMPLATE.xml" h6. Creating a series of homedirs with metadata and
permissions
Template file
```

```
createSpace -path="/files" -ldap="((&(type=pers)(department=123))(uid=55*))" -ldapAttribute="uid" -templateFile="/usr/local/src/templates/homedirs.xml" -
regexp="/(files)/(:).:/files(/:).:/files(/:).:/files(/:).:/files(/:).:/files(/:).:/files(/:).:/files(/:).:/files(/:).:/files(/:).:/files(/:).:/files(/:).:/files(/:)." -regexpSeparator=":" -forceSetProperties="true"
```

Consequences of this command :

- "uid" attributes matching the given LDAP filter are retrieved from the LDAP
- The regexp is applied to each "uid" giving hashed paths like "t/tb/tbellemb"
- The "path" is appended to each hashed path giving the final path like "/files/t/tb/tbellemb"



The v4.2 storage channel requires to set specific permissions and metadata on homedirs. More precisely, the channels relies on the owner metadata to manage permissions. Whether you already have or not homedirs on you WebDAV server, you can use the `createSpace` command with the

parameter set to true to set right properties and permissions. Use the following pattern :


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