CONVERSATIONS

Command Line

Version 2.7

VIPcom
Communications

VIPcom GmbH
Ruedesheimer Strasse 7
80686 Munich
Phone.: +49 89 54750-0
Fax: +49 89 54750-200
e-mail: info@vipcomag.de
http://www.vipcomag.de

The use, reproduction or distribution of the programme is subject to the restrictions described in your contract with VIPcom GmbH
The specifications contained in this book are subject to correction and may be altered without any further notification. VIPcom GmbH does not thereby commit itself to any further obligations.

Introduction

This manual is written for the administrator on how to use the Command Line Interface Tool umapi-admin. With this tool conversations (formerly teamXchange) can be administered via the command level.

Further Documentation

- Creating a User Workstation (installing MAPI connector and creating a profile)
- Administration
- Installing and Configuring Apache Server on Windows
- User Manual

Content

A. umapi-admin - Administration commands

Via the umapi-admin commands you can execute administrator actions on the command level, e.g. "enter users in the database" or "carry out LDAP queries".

umapi-admin is installed along with **conversations**.
You will find it
under on Unix/Linux in the directory: `/usr/local/umapi/bin`
under on Windows in the directory: `c:\Programme\conversations\bin`

Starting umapi-admin:

- Enter the path to `umapi-admin.exe`.

For Windows:

```
c:\Programme\conversations\bin\umapi-admin.exe
```

For Unix/Linux you have to enter `umapi-admin` if you are logged in as user `conversations`

1. Syntax

You always have to indicate the command `umapi-admin`.
umapi-admin equires three arguments:

arg1: command name
arg2: name of the computer where **conversations** is installed or: `localhost`
arg3: administrator password

If no administrator password exists, enter `""`

Parameters

Behind argument 3 parameters can be entered.

`umapi-admin` *arg1 arg2 arg3 parameter1 parameter2*

Example 1:

```
umapi-admin  UserGet  localhost  ""  mick.baxter
```

displays the data of the user `mick.baxter`

Output:

```
mick.baxter
Mick Baxter
@
```

Any output always will be completed with an @-sign.

Input

Some of the commands require input:

Example 2:

```
umapi-admin UserSetPassword localhost ""
mick.baxter
sunshine
@
```

The example sets the password `sunshine` for the user `mick.baxter`

Any output always will be completed with an @-sign

2. Command Overview

Administrator	Command
To **set** or to **change** the administrator password	AdmSetPassword 9
To **read** the license file	LicensePut 9

User	Command
To **register** a new user account	UserCreate 10
To **enter** or to **change** a user description	UserPut 11
To **set** a user password	UserSetPassword 11
To **display** the user data	UserGet 12
To **display** the user list	UserGetList 12
To **delete** a user account	UserDelete 13
To **assigne** a user to a group	UserAddGroup 13
To **remove** a user from the group list	UserRemGroup 13
To **display** the group membership of a user	UserGetGroupList 14

Group	Command
To **register** a new group	GroupCreate 15
To **enter** or change a group description	GroupPut 15
To **display** the group data	GroupGet 16
To **display** the group list	GroupGetList 16
To **display** the group members	GroupGetMembers 16
To **delete** a group	GroupDelete 17

Folder	Command
To **display** the folder list	`FolderGetList` 18
To **set** folder permissions	`FolderPutAcl` 21
To **display** folder permissions	`FolderGetAcl` 19
To **display** folder permissions of a user or a group	`FolderGetRights` 23

Configuration	Command
To **display** a configuration value	`ConfigGet` 24
To **set** a configuration value	`ConfigPut` 27
To **delete** a configuration value	`ConfigDel` 28
	`ConfigGetCategories`
	`ConfigGetCategoryVars`

3. Administrator Commands

AdmSetPassword

Sets or changes the administrator password.

`umapi-admin AdmSetPassword host` *old_password new_password*

@

Parameters

old_password:	The old password must be known. Otherwise an error message is displayed. Exception: If no password is set yet, `""` is valid.
new_password:	The new password.

LicensePut

Adds the license data to the database.

`umapi-admin LicensePut`

license text

@

Parameter

license text	License key (Text from `BEGIN: LICENSE` to `END: LICENSE)`

4. User Commands

Notice:

Commands to enter user data in the database (`UserCreate, UserPut, UserSetPassword`)are only effective if **no** LDAP database is used.

`UserGetList` and `UserDelete` also can be used with LDAP.

UserCreate

adds a new user account to the database.

`umapi-admin UserCreate` ***host password***

userid

@

Parameter

userid	Name of the user. This user name becomes part of the e-mail address. It will be the prefix to the "@" sign of the users e-mail-address. Therefore it is recommended to use the user's first and last name, separated with a dot. (No spaces or special characters. Only lower case letters may be used.)

Example

```
umapi-admin UserCreate myhost passwort
mick.baxter
@
```

No output. If the entry exists already: error code 31 (see below).

UserPut

Enters or changes a user description.

`umapi-admin UserPut` ***host password***

usererid

descripion

@

Parameters

userid	Name of the user.
description	description (max. 254 characters).

UserSetPassword

Sets a password for a user..

`umapi-admin UserSetPassword` ***host password***

usererid

userpasswort

@

Parameters

userid	Name of the user.
userpasswort	Password of the user.

UserGet

Displays the data of a user.

`umapi-admin UserGet` ***host password benutzerid***

Parameter

usererid	Name of the user.

Output

usererid

descripion

@

UserGetList

Displays the data of all users.

`umapi-admin UserGetList` ***host password***

Output

userid1

descripion1

@

userid2

descripion2

@

UserDelete

Deletes a user from the database.

`umapi-admin UserDelete` ***host password***

userid

@

Parameter

userid	Name of the user (see `UserCreate`).

UserAddGroup

Assigns a user to a group.

`umapi-admin UserAddGroup` ***host password***

userid

groupid

@

Parameters

userid	Name of the user.
groupid	Name of the group.

UserRemGroup

Removes a user from a group.

`umapi-admin UserRemGroup` ***host password***

userid

groupid

@

Parameters

userid	Name of the user.
groupid	Name of the user.

UserGetGroupList

Displays the group membership of a user.

`umapi-admin UserGetGroupList` ***host password userid***

Parameter

userid	Name of the user

Output

usergroup1

@

usergroup2

@

...

5. Group Commands

GroupCreate

Registers a new group.

You can set folder permissions for groups exactly as for users.

`umapi-admin GroupCreate` ***host password***

groupid

@

Parameter

groupid	Name of the group. (No spaces or special characters. Only lower case letters may be used.)

Example

```
umapi-admin GroupCreate myhost passwort
project team1
@
```

No output. If this entry exists already: error code.

GroupPut

Enters or changes a group description.

`umapi-admin GroupPut` ***host password***

groupid

description

@

Parameters

groupid	Name of the group.
description	Description (max. 254 characters).

GroupGet

Displays the data of a group.

`umapi-admin GroupGet` ***host password groupid***

Parameter

groupid	Name of the group.

Output

groupid

description

@

GroupGetList

Displays a list of all user groups.

`umapi-admin GroupGetList` ***host password***

Output

groupid1

description1

@

groupid2

description2

@

...

GroupGetMembers

Displays all members of a user group.

```
umapi-admin GroupGetMembers host password groupid
```

groupid	Name of the group.

Output

userid1

description1

@

userid1

description2

@

...

GroupDelete

Deletes the entire group from the database.

➢ Before a group can be deleted all corresponding users have to be removed using the command `UserRemGroup` (see above UserRemGroup 13)

```
umapi-admin GroupDelete host password
groupid
@
```

Parameter

groupid	Name of the group.

6. Folder Commands

The term "Folders" is used for contact folders, tasks or calenders.

- Permissions are only assigned to folders not to documents, documents inherit the rights of their mother folder.
- Subfolders, which are created later on, automatically inherit the rights of the "mother"-folder.
- Group permissions always take priority over user permissions. See documentation " **conversations** - Administration", "Assigning Rights".

FolderGetList

Displays the subfolder structure of a folder.

`umapi-admin FolderGetList` ***host password folder***

folder	*folder* means: • area `public` or `users`. • the path to the folder of a user (*entryIdString*).

Output

entryIdString	(Folder identity string as a text)
name	*(*Name of the user who has folder access*)*
subfolder	*(*`1`, if the folder contains subfolders, `0`, if the folder contains subfolders).

@

Example 1 - Public:

To display the structure of the area `public.`

Input

```
umapi-admin FolderGetList myserver mypassword public
```

Output

```
900b1726c7a64b6ba8d563a3f9050a2b
Public
```

```
1
@
```

Example 2 - Public:

To display the structure of the folder `All public folders.`

Input

```
umapi-admin FolderGetList myserver mypassword
900b1726c7a64b6ba8d563a3f9050a2b
```

Output

```
b8a686d7518b470d8e7e100238f7fd5b
Calendars
0
@

b7fcc38e163a4eb5a1673438f1e65651
Employees
1
@
```

FolderGetAcl

Displays folder permissions.

`umapi-admin FolderGetAcl` ***host password entryIdString***

Parameter

entryIdString	Folder identity string. You get the *entryIdString* using the command `FolderGetList` 18.

Output

entryIdString

userid/groupid

type

mask

@

entryIdString	Folder identity string (repeat).
userid/groupid	Name of the user or the group.
type	`0` = user permissions, `1` = group permissions.
mask	Hexadecimal output of the permissions (see below).

Description of the Access Rights

position	hexadezimal value	access right
0	1	create message
1	2	read message
2	4	write message
4	10	delete message
6	40	create folder
7	80	read folder
8	100	write folder
10	400	delete folder

The addition of the access rights will be displayed in hexadecimal values.

If all permissions are set, the output will be: 5d7 (400+100+80+40+10+4+2+1)
Added with the disengaged positions (e. g. Public folder), the output will be: f f f f f f f f

Example

Input

```
umapi-admin FolderGetAcl myserver mypassword
b7fcc38e163a4eb5a1673438f1e65651
```

Output

```
b7fcc38e163a4eb5a1673438f1e65651
mick.baxter
0
```

3

@

sales

1

5d7

@

Comment: The user `Mick Baxter` has got the permissions `create` and `read` messages for the folder b7fcc38e163a4eb5a1673438f1e65651 and its subfolders. The group `sales` has got the permissions `read, create, change` and `delete` for subfolders as well as messages.

FolderPutAcl

Sets permissions to the folder.

`umapi-admin FolderPutAcl` ***host password entryIdString***

userid/groupid

type

mask

@

Parameters

entryIdString	Folder identity string. You get the entryIdString with the command FolderGetList 18.
userid/groupid	Name of the user or the group.
type	0 = user permissions, 1 = group permissions.
mask	Addition of the hexadecimal values of the access rights (see above Description of the access rights 20).

Example

```
umapi-admin FolderPutAcl myserver mypassword
b7fcc38e163a4eb5a1673438f1e65651
mick.baxter
0
123
@
sales
1
5d3
@
```

Comment: The user `Mick Baxter` and the group `sales` (see above `FolderGetAcl` 19) have got other permissions then in the example before: In addition to his rights (to read and create messages) `Mick Baxter` gets the rights to create sufolders and read their content (1+2+40+80=123).
The group `sales` does not have the right to create subfolders (5d7 - 40 = 597) any longer.

FolderGetRights

Displays the folder permissions of a user or a group.

`umapi-admin FolderGetRights` ***host password entryIdString user***

Parameter

entryIdString	You get the *entryIdString* using the command `FolderGetList` 18.
user	user or group.

Output

entryIdString

user

mask (hexadecimal output of the access rights)

@

7. Configuration Commands

Configuration parameters are set in the database during the installation.

With the configuration commands you can read, change, set and delete parameters.

ConfigGet

Displays the value of a configuration parameter.

`umapi-admin ConfigGet` ***host password kategorie id***

category	stands for a certain category of the configuration data. Current categories: `SERVER`, `SERVER.LDAP`, `SERVER.LDAP.AUTH`
id	stands for one parameter of the category. Description see below.

Parameters of the Category SERVER

ID	Description	Value
`ACLURL`	URL to the **conversations** Rights Management, which is configured in the Tomcat sever.	*URL*
`DOMAIN`	domain name of the server where **conversations** is installed.	*domain name.* Example: `txc.vipcomag.de`
`PORT1`	1st port used by the server*.	`8000`
`TRACELEVEL`	trace level (see below `TraceSetLevel` 28)	Example: `0`
`TRACEFLAGS`	trace flags	Example: `65664`
`NORMALPROPLIMIT`	The maximum size for an entry in an attribute field in the database.	Example: `256`

*Port 1 can be changed. Port 2 is always one digit higher than port 1.

Caution: The port should not be changed if possible!

☞ If you do not use 8000 for port 1 you have to consider the following:

If the host name of the **conversations** server is required, you always will have to fill in also the port number.
This concerns the login for the administration program as well as all Outlook profiles and
umapi admin interface commands.

Instead of just the hostname you will have to enter: *hostname*:*port*, e.g.: `localhost:4711`

ID Values of the Category SERVER.LDAP (LDAP server access)

ID	Description	Value
`URI`	Address of the LDAP server.	*ldap://hostname:port* Example: `ldap://ldaphost.vipcom.de:389`
`USER`	Login account for access to the LDAP server, if an anonymous access is not possible.	*user name*
`PASSWD`	The user's password for the login account.	*password*
`REFFERALS`	Follow the recommendations to other directories or not.	`0` = do not follow `1` = follow
`DGBLDAP`	Debug messages in the logbook	`0` = no messages (default) `-1` = messages)
`DGBLBER`	Debug messages in the logbook	`0` = no messages (default) `-1` = messages)

ID Values of the Category SERVER.LDAP.AUTH (LDAP server request)

ID	Description	Value
`USER.DN`	LDAP directory, in which the query for the user shall start.	*path/directory* Example see below [*1]
`USER.SCOPE`	Set search scope.	`1` or `2`

	1 = one sublevel in addition 2 = all sublevels	Default: 2
`USER.LOGINATTR`	describes the database field'sname of the login attribute.	Example: `samAccountName`
`USER.MAILATTR`	describes the database field'sname of the email address.	Example: `mail`
`USER.DISPLAYATTR`	describes the database field's name of the user's short name, which is shown at email dispatch.	Example: `displayName`
`GROUP.DN`	describes the database fields name of a user group, to which all **conversations** users must belong.	*path/directory* Example see below [*2]
`GROUP.MEMBERATTR`	describes the database fields name of the group members.	Example: `member`

[*1] OU=Munich,DC=vipcomag,DC=de

[*2] CN=conversations,OU=Munich,DC=vipcomag,DC=de

ID Values of the Category MIMESND (SMTP sending messages)

Parameters	Description	Value
JDEBUG	enables Javamail debug	1 = enables Javamail debug
SMTPHOST	If a smart host shall take over sending, enter the name of this host here.	*host name* Default: `localhost`
UMSGW	If a PAPAGENO fax server shall be used enter the name of the PAPAGENO Gateway server here.	*host name*

ID Values of the Category MIMERCV (SMTP - receiving messages)

Parameters	Description	Value
KEEPLOG	Keeps all incoming emails in the log directory (undeliverable incoming emails with parse error will not be saved)	1 = keep all
LOGIO	Save undeliverable incoming emails before parse. (Thus all undeliverable emails can be saved).	1 = save

Output

category

id

value

@

Example

```
umapi-admin ConfigGet localhost mypasswd SERVER DOMAIN
```

Output

```
SERVER
DOMAIN
conversations
@
```

ConfigPut

Sets a configuration value.

`umapi-admin ConfigPut` ***host password***

category

id

value

@

category and *id* are described above under `ConfigGet` 24. *Value* is the input-value.

`Category SERVER:`

The `TRACELEVEL` is set using `TraceSetLevel` (see below).

Example

The following input

```
umapi-admin ConfigPut localhost mypasswd
SERVER
DOMAIN
conversations
@
```

sets the domain `conversations`.

ConfigDel

Deletes a configuration value from the database.

`umapi-admin ConfigDel` ***host password***

category

id

@

category and *id* are described above under `ConfigGet` 24

TraceSetLevel

Sets an output level.

`umapi-admin TraceSetLevel` ***host password level***

Level	Description
0	Errors

1	Warnings
2	Debug
3	Call (records all function calls)

Attention! Trace level produces numerous output which may cause the **conversations** server to slow down.

Trace flags can be placed only if trace level 3 is set.

In order to trace errors efficiently contact our support team!

B. Error Code

0	Ok
1	Error
2	Invalid parameter(s)
3	RPC connection failed
4	RPC call failed
5	Entry already exits
6	Entry does not exist
7	False password
8	No cache memory
9	Not implemented
10	Invalid license
11	License limit too large

Further information on errors can be possibly obtained in stderr.

Index

Index

Index

VIPcom
Communications