

# Manage recipients history WebApp

These scripts and tools are provided to help you manage your Kopano system. They are provided as a free service for Kopano users, without warranty of any kind.

Please be aware that usage is at your own risk!

## manage\_recipients.py

master KSC/webapp-tools

```
#!/usr/bin/env python
#encoding: utf-8

import kopano
from MAPI.Util import *
import json

def opt_args():
    parser = kopano.parser('skpcfm')
    parser.add_option("--user", dest="user", action="store", help="Run script for user")
    parser.add_option("--list", dest="list", action="store_true", help="List recipients history")
    parser.add_option("--backup", dest="backup", action="store_true", help="Backup recipients history")
    parser.add_option("--restore", dest="restore", action="store_true", help="Restore recipients history")
    parser.add_option("--restore-file", dest="restorefile", action="store", help="Restore from an other file then username.json")
    parser.add_option("--remove", dest="remove", action="store", help="Remove recipients ")
    parser.add_option("--remove-all", dest="removeall", action="store_true", help="Remove complete recipients history")
    parser.add_option("--dry-run", dest="dryrun", action="store_true", help="Test script")

    return parser.parse_args()

def main():
    options, args = opt_args()

    if not options.user:
        print 'Please use:\n %s --user <username>' % (sys.argv[0])
        sys.exit(0)

    user = kopano.Server(options).user(options.user)
    webapp = user.store.prop(0X6773001F).value
    webapp = json.loads(webapp)

    if options.backup:
        f = open('%s.json' % user.name, 'w')
        f.write(json.dumps(webapp, sort_keys=True,
```

```

        indent=4, separators=(',', ' : ')))
f.close()
sys.exit(0)

if options.restore:
    if options.restorefile:
        filename = options.restorefile
    else:
        filename = '%s.json' % user.name
    with open(filename) as data_file:
        data = json.load(data_file)
    user.store.mapiobj.SetProps([SPropValue(0X6773001F, u'%s' % json.dumps
(data))])
    user.store.mapiobj.SaveChanges(KEEP_OPEN_READWRITE)
    sys.exit(0)

if options.list:
    print json.dumps(webapp, sort_keys=True,
        indent=4, separators=(',', ' : '))
    sys.exit(0)

if options.remove:
    newlist = json.loads('{"recipients":[]}')
    for rec in webapp['recipients']:
        if options.remove in rec['display_name'] or options.remove in rec
['smtp_address'] \
            or options.remove in rec['email_address']:
            print 'removing contact %s [%s]' % (rec['display_name'], rec
['smtp_address'])
        else:
            newlist['recipients'].append(rec)

    if not options.dryrun:
        user.store.mapiobj.SetProps([SPropValue(0X6773001F, u'%s' % json.dumps
(newlist))])
        user.store.mapiobj.SaveChanges(KEEP_OPEN_READWRITE)

    sys.exit(0)

if options.removeall:
    newlist = json.loads('{"recipients":[]}')
    if not options.dryrun:
        user.store.mapiobj.SetProps([SPropValue(0X6773001F, u'%s' % json.dumps
(newlist))])
        user.store.mapiobj.SaveChanges(KEEP_OPEN_READWRITE)

    sys.exit(0)

if __name__ == "__main__":
    main()

```

Usage:

Usage: manage\_recipients.py [options]

Options:

-h, --help	show this help message and exit
-c FILE, --config=FILE	load settings from FILE
-s SOCKET, --server-socket=SOCKET	connect to server SOCKET
-k FILE, --ssl-key=FILE	SSL key file
-p PASS, --ssl-pass=PASS	SSL key password
-f NAME, --folder=NAME	run program for specific folder
-m, --modify	enable database modification
--user=USER	Run script for user
--list	List recipients history
--backup	Backup recipients history
--restore	Restore recipients history
--restore-file=RESTOREFILE	Restore from an other file then username.json
--remove=REMOVE	Remove recipients
--remove-all	Remove complete recipients history
--dry-run	Test script

## List recipients

```
python manage_recipients.py --user <user> --list
```

## Remove recipient

Remove option is searching in display\_name, smtp\_address or email\_address.

```
python manage_recipients.py --user <user> --remove <recipient name>
```

## Clear history

```
python manage_recipients.py --user <user> --remove-all
```

## Example

Remove all recipients that have example.com in there display\_name, smtp\_address or email\_address

```
python manage_recipients.py --user user --remove example.com
```