



grommunio

grommunio Man Pages

Release 2021.10.06

grommunio GmbH

Oct 06, 2021

CONTENTS

1	gromox	1
1.1	Name	1
1.2	Description	1
1.3	Configuration files	1
1.4	Databases	1
1.5	Definitions	2
1.6	Exchange subsystem	2
1.7	PHP-MAPI subsystem	2
1.8	Mail retrieval agent subsystem	2
1.9	Local delivery agent	3
1.10	Auxiliary services	3
1.11	System administration	3
1.12	Service plugins	4
1.13	Past commands that have been removed	4
1.14	Language bindings	4
1.15	Listening sockets	4
1.16	Files	5
2	grommunio-admin	6
2.1	Name	6
2.2	Synopsis	6
2.3	Description	6
2.4	Commands	6
2.4.1	config	6
2.4.2	connect	7
2.4.3	dbconf	7
2.4.4	domain	7
2.4.5	fetchmail	7
2.4.6	fs	7
2.4.7	ldap	7
2.4.8	mconf	7
2.4.9	mlist	7
2.4.10	passwd	7
2.4.11	run	7
2.4.12	service	8
2.4.13	shell	8
2.4.14	taginfo	8
2.4.15	user	8
2.4.16	version	8
2.5	See Also	8
3	grommunio-admin-config	9
3.1	Name	9
3.2	Synopsis	9

3.3	Commands	9
3.3.1	check	9
3.3.2	dump, get	9
3.3.3	trace	10
3.4	Options	10
3.5	Tracing	10
3.5.1	By-File	10
3.5.2	By-Value	10
3.6	See Also	11
4	grommunio-admin-connect	12
4.1	Name	12
4.2	Synopsis	12
4.3	Description	12
4.4	Options	12
4.5	Filesystem Emulation	13
4.6	See Also	13
5	grommunio-admin-dbconf	14
5.1	Name	14
5.2	Synopsis	14
5.3	Description	14
5.4	Commands	14
5.5	Options	15
5.6	Commit Hooks	15
5.7	Available Commit Commands	15
5.7.1	Key	15
5.7.2	File	15
5.7.3	Service	15
5.8	Command Variable Expansion	16
5.9	See Also	16
6	grommunio-admin-domain	17
6.1	Name	17
6.2	Synopsis	17
6.3	Description	17
6.4	Commands	17
6.5	Options	18
6.6	Fields	18
6.7	See Also	18
7	grommunio-admin-fetchmail	19
7.1	Name	19
7.2	Synopsis	19
7.3	Description	19
7.4	Commands	19
7.5	Options	20
7.6	Fields	20
7.7	See Also	21
8	grommunio-admin-fs	22
8.1	Name	22
8.2	Synopsis	22
8.3	Description	22
8.4	Commands	22
8.5	Options	22
8.6	See Also	23
9	grommunio-admin-ldap	24

9.1	Name	24
9.2	Synopsis	24
9.3	Description	24
9.4	Commands	24
9.5	Options	25
9.6	See Also	25
10	grommunio-admin-mconf	26
10.1	Name	26
10.2	Synopsis	26
10.3	Description	26
10.4	Commands	26
10.5	Options	27
10.6	See Also	27
11	grommunio-admin-mlist	28
11.1	Name	28
11.2	Synopsis	28
11.3	Description	28
11.4	Commands	28
11.5	Options	29
11.6	See Also	29
12	grommunio-admin-passwd	30
12.1	Name	30
12.2	Synopsis	30
12.3	Description	30
12.4	Options	30
12.5	See Also	30
13	grommunio-admin-run	31
13.1	Name	31
13.2	Synopsis	31
13.3	Description	31
13.4	Options	31
13.5	See Also	31
14	grommunio-admin-service	32
14.1	Name	32
14.2	Synopsis	32
14.3	Description	32
14.4	Commands	33
	14.4.1 disable	33
	14.4.2 load	33
	14.4.3 status	33
14.5	Options	33
14.6	Services	33
14.7	See Also	33
15	grommunio-admin-shell	34
15.1	Name	34
15.2	Synopsis	34
15.3	Description	34
15.4	Options	34
15.5	See Also	34
16	grommunio-admin-taginfo	35
16.1	Name	35
16.2	Synopsis	35

16.3	Description	35
16.4	Options	35
16.5	See Also	35
17	grommunio-admin-user	36
17.1	Name	36
17.2	Synopsis	36
17.3	Description	36
17.4	Commands	36
17.5	Options	36
17.6	See Also	37
18	grommunio-admin-version	38
18.1	Name	38
18.2	Synopsis	38
18.3	Description	38
18.4	Options	38
18.5	See Also	38
19	grommunio-dbconf	39
19.1	Name	39
19.2	Synopsis	39
19.3	Description	39
19.4	Commands	39
19.5	Options	40
19.6	Commit Hooks	40
19.7	Available Commit Commands	40
	19.7.1 Key	40
	19.7.2 File	40
	19.7.3 Service	41
19.8	Command Variable Expansion	41
19.9	See Also	41
20	adaptor	42
20.1	Name	42
20.2	Synopsis	42
20.3	Description	42
20.4	Options	42
20.5	Files	42
20.6	Configuration file directives	43
20.7	See also	43
21	alias_translator	44
21.1	Name	44
21.2	Description	44
21.3	See also	44
22	at_client	45
22.1	Name	45
22.2	Description	45
22.3	Configuration file directives	45
22.4	Caveats	45
22.5	See also	45
23	authmgr	46
23.1	Name	46
23.2	Description	46
23.3	Configuration file directives	46
23.4	Authentication modes	46

23.5 See also	47
24 autodiscover	48
24.1 Name	48
24.2 Description	48
24.3 Configuration file directives	48
24.3.1 [database]	49
24.3.2 [exchange]	49
24.3.3 [default]	49
24.3.4 [system]	49
24.3.5 [http-proxy]	49
24.4 Normative references	49
24.5 See also	50
25 delivery-queue	51
25.1 Name	51
25.2 Synopsis	51
25.3 Description	51
25.4 Options	51
25.5 Configuration file directives	51
25.6 Files	54
25.7 See also	54
26 delivery	55
26.1 Name	55
26.2 Synopsis	55
26.3 Options	55
26.4 Plugins	55
26.5 Configuration file directives	55
26.6 Files	57
26.7 See also	57
27 str_table	58
27.1 Name	58
27.2 Description	58
27.3 Configuration file directives	58
27.4 Files	59
27.5 See also	59
28 event	60
28.1 Name	60
28.2 Synopsis	60
28.3 Description	60
28.4 Options	60
28.5 Files	60
28.6 Configuration file directives	61
28.7 Event protocol	61
28.8 See also	62
29 event_proxy	63
29.1 Name	63
29.2 Description	63
29.3 See also	63
30 event_stub	64
30.1 Name	64
30.2 Description	64
30.3 See also	64

31 exchange_emsmb	65
31.1 Name	65
31.2 Description	65
31.3 Config file directives	65
31.4 Files	66
31.5 Normative references	66
31.6 See also	66
32 exchange_nsp	67
32.1 Name	67
32.2 Description	67
32.3 Config file directives	67
32.4 Notes	67
32.5 Normative references	68
32.6 See also	68
33 exchange_rfr	69
33.1 Name	69
33.2 Description	69
33.3 Configuration file directives	69
33.4 Normative references	69
33.5 See also	69
34 exmdb_local	70
34.1 Name	70
34.2 Description	70
34.3 Config file directives	70
34.4 Files	70
34.5 See also	70
35 exmdb_provider	71
35.1 Name	71
35.2 Description	71
35.3 Configuration file directives	71
35.4 Multiserver selection map	72
35.5 Network protocol	73
35.6 Files	73
35.7 See also	73
36 freebusy	74
36.1 Name	74
36.2 Description	74
36.3 Files	74
36.4 See also	74
37 gromox-abktconv	75
37.1 Name	75
37.2 Synopsis	75
37.3 Description	75
37.4 Options	75
37.5 Normative references	75
37.6 See also	76
38 gromox-abktpull	77
38.1 Name	77
38.2 Synopsis	77
38.3 Description	77
38.4 Normative references	77
38.5 See also	77

39 gromox-dbop	78
39.1 Name	78
39.2 Synopsis	78
39.3 Options	78
39.4 See also	78
40 gromox-kdb2mt	79
40.1 Name	79
40.2 Synopsis	79
40.3 Description	79
40.4 Options	79
40.5 Environment variables	80
40.6 Examples	80
40.7 See also	80
41 gromox-mailq	81
41.1 Name	81
41.2 Synopsis	81
41.3 Description	81
41.4 See also	81
42 gromox-mkmidb	82
42.1 Name	82
42.2 Synopsis	82
42.3 Options	82
42.4 Files	82
42.5 See also	82
43 gromox-mkprivate	83
43.1 Name	83
43.2 Synopsis	83
43.3 Options	83
43.4 Files	83
43.5 Config file directives	83
43.6 See also	84
44 gromox-mkpublic	85
44.1 Name	85
44.2 Synopsis	85
44.3 Options	85
44.4 Files	85
44.5 Config file directives	85
44.6 See also	86
45 gromox-mt2exm	87
45.1 Name	87
45.2 Synopsis	87
45.3 Description	87
45.4 Options	87
45.5 Examples	87
45.6 See also	88
46 gromox-pff2mt	89
46.1 Name	89
46.2 Synopsis	89
46.3 Description	89
46.4 Options	89
46.5 Examples	90
46.6 See also	90

47 gromox	91
47.1 Name	91
47.2 Synopsis	91
47.3 Description	91
47.4 See also	91
48 gromox-rebuild	92
48.1 Name	92
48.2 Synopsis	92
48.3 Options	92
48.4 Files	92
48.5 Config file directives	92
48.6 See also	92
49 http	93
49.1 Name	93
49.2 Synopsis	93
49.3 Description	93
49.4 Options	93
49.5 URI processing order	93
49.6 RPC-over-HTTP	94
49.7 Configuration file directives	94
49.8 Files	97
49.9 Normative references	97
49.10 See also	97
50 imap	98
50.1 Name	98
50.2 Synopsis	98
50.3 Options	98
50.4 Configuration file directives	98
50.5 Files	101
50.6 See also	101
51 ip6_container	102
51.1 Name	102
51.2 Description	102
51.3 Configuration file directives	102
51.4 See also	102
52 ldap_adaptor	103
52.1 Name	103
52.2 Description	103
52.3 Configuration file directives	103
52.4 See also	104
53 logthru	105
53.1 Name	105
53.2 Description	105
53.3 Configuration file directives	105
53.4 See also	105
54 mapi	106
54.1 Name	106
54.2 Description	106
54.3 Configuration	106
54.4 Compatibility with other implementations	106
54.5 See also	107

55 mapi	108
55.1 Name	108
55.2 Definition	108
55.3 Gromox implementation	109
56 message_enqueue	110
56.1 Name	110
56.2 Description	110
56.3 See also	110
57 mh_emsmbd	111
57.1 Name	111
57.2 Description	111
57.3 Config file directives	111
57.4 Normative references	111
57.5 See also	111
58 mh_nsp	112
58.1 Name	112
58.2 Description	112
58.3 Config file directives	112
58.4 Normative references	112
58.5 See also	112
59 midb	113
59.1 Name	113
59.2 Synopsis	113
59.3 Description	113
59.4 Options	113
59.5 Configuration file directives	113
59.6 Files	115
59.7 See also	115
60 midb_agent	116
60.1 Name	116
60.2 Configuration file directives	116
60.3 Multiserver map	116
60.4 Files	117
60.5 See also	117
61 mod_cache	118
61.1 Name	118
61.2 Description	118
61.3 Config file directives	118
61.4 URI map	118
61.4.1 Default entries	118
61.5 Files	119
61.6 See also	119
62 mod_fastcgi	120
62.1 Name	120
62.2 Description	120
62.3 Config file directives	120
62.4 URI map	120
62.4.1 Default entries	121
62.5 Files	121
62.6 See also	121
63 mod_rewrite	122

63.1	Name	122
63.2	Description	122
63.3	Config file directives	122
63.4	Rewrite map	122
63.5	Default rules	122
63.6	Files	123
63.7	See also	123
64	mysql_adaptor	124
64.1	Name	124
64.2	Description	124
64.3	Configuration file directives	124
64.4	See also	125
65	pam_gromox	126
65.1	Name	126
65.2	Description	126
65.3	Incantation in /etc/pam.d/smtp	126
65.4	Configuration directives in /etc/gromox/pam.cfg	126
65.5	See also	127
66	pop3	128
66.1	Name	128
66.2	Synopsis	128
66.3	Options	128
66.4	Configuration file directives	128
66.5	Files	130
66.6	See also	130
67	rtf2html	131
67.1	Name	131
67.2	Description	131
67.3	Options	131
67.4	Files	131
67.5	Normative references	131
67.6	See also	131
68	sa.cfg	132
68.1	Name	132
68.2	Directives	132
68.3	See also	133
69	str_table	134
69.1	Name	134
69.2	Description	134
69.3	Configuration file directives	134
69.4	Files	135
69.5	See also	135
70	textmaps	136
70.1	Name	136
70.2	Description	136
70.3	Files	136
70.4	See also	136
71	timer	137
71.1	Name	137
71.2	Synopsis	137
71.3	Description	137

71.4	Options	137
71.5	Files	137
71.6	Configuration file directives	138
71.7	Timer protocol	138
71.8	See also	138
72	timer_agent	139
72.1	Name	139
72.2	Description	139
72.3	Configuration file directives	139
72.4	See also	139
73	filter	140
73.1	Name	140
73.2	Configuration file directives	140
73.3	Files	141
73.4	See also	141
74	zcore	142
74.1	Name	142
74.2	Synopsis	142
74.3	Description	142
74.4	Options	142
74.5	Network protocol	142
74.6	Store lookup	143
74.7	Files	143
74.8	Configuration file directives	143
74.9	Files	145
74.10	Notes	145
74.11	See also	145
75	Legal Notice	146

1.1 Name

gromox — overview of the Gromox groupware server

1.2 Description

Gromox is a groupware server capable of serving as a replacement for Microsoft Exchange. Connectivity options include RPC/HTTP (Outlook Anywhere), IMAP, POP3, an SMTP-speaking LDA, and a PHP module with a Z-MAPI function subset.

Gromox relies on other components to provide a sensibly complete mail system, such as Postfix as a mail transfer agent, and grommunio-admin for user management. A web interface is available with grommunio-web. The grommunio distribution ships these essentials and has a ready-to-run installation of Gromox. system.

1.3 Configuration files

Program configuration files reside within `/etc/gromox`. Most of the programs have a `config_file_path` directive with which the search path for further config files can be specified. For example, `http(8gx)` defaults to `config_file_path=/etc/gromox/http:/etc/gromox`, so the `mysql_adaptor(4gx)` plugin as loaded by `http` will first try `/etc/gromox/http/mysql_adaptor.cfg`, then `/etc/gromox/mysql_adaptor.cfg`. This allows having one file that is shared between multiple programs as well as being able to override on a per program-basis.

1.4 Databases

- User information is held in a MariaDB/MySQL database. This database can be accessed by multiple Gromox servers, and so enables distributed Gromox operation. The MariaDB system itself provides the necessary utilities for distributing or replicating this database.
- Per-user e-mail messages are stored in a SQLite database (e.g. `/var/lib/gromox/user/m1/1/1/exchange.sqlite3`), as is a message index (e.g. `/var/lib/gromox/user/m1/1/1/midb.sqlite3`). These are normally only used by one system, but can be shared through network filesystems provided that file locking is properly implemented in the filesystem driver. Normal file mechanisms can be used to backup or transfer the database to another Gromox host.

1.5 Definitions

- `mapi(7gx)` — Definition for “Messaging Application Programming Interface”

1.6 Exchange subsystem

- `autodiscover(4gx)` — Autodiscover HTTP Service Protocol handler
- `exchange_emsmdb(4gx)` — `http(8gx)` processing plugin for the Wire Format Protocol (Outlook/Exchange RPCs).
- `exchange_nsp(4gx)` — `http(8gx)` processing plugin for the Exchange Server Name Service Provider Interface Protocol.
- `exchange_rfr(4gx)` — `http(8gx)` processing plugin for the Address Book Name Service Provider Interface Referral Protocol.
- `exmdb_provider(4gx)` — `http(8gx)` service plugin for `exmdb` connections
- `freebusy(8gx)` — Helper program for the EWS Freebusy mechanism
- `http(8gx)`
- `mod_cache(4gx)` — `http(8gx)` processing plugin for serving objects from a local filesystem
- `mod_fastcgi(4gx)` — `http(8gx)` processing plugin for proxying requests to FastCGI servers
- `mod_rewrite(4gx)` — `http(8gx)` processing plugin for altering HTTP request URIs before processing
- `mh_emsmdb(4gx)` — MAPI-HTTP processor for EMSMDB
- `mh_nsp(4gx)` — MAPI-HTTP processor for NSPI
- `timer(8gx)` — deferred command executor

1.7 PHP-MAPI subsystem

- `rtf2html(1gx)` — C++ helper program for the `mapi(4gx)` `mapi_decompressrtf` function
- `zcore(8gx)` — Bridge for PHP-MAPI requests

1.8 Mail retrieval agent subsystem

- `imap(8gx)` — IMAP server
- `midb_agent(4gx)` — Service plugin for conversing with `midb(8gx)`
- `pop3(8gx)` — POP3 server

1.9 Local delivery agent

- `alias_translator(4gx)` — Alias resolution plugin for `delivery(8gx)`
- `delivery(8gx)` — Backend for local delivery
- `delivery-queue(8gx)` — LMTP/SMTP frontend for local delivery
- `domain_list(4gx)` — instance of `str_table(4gx)`
- `message_enqueue(4gx)` — Message enqueueing flusher plugin for `delivery-queue(8gx)`
- `str_table(4gx)` — string table match plugin
- `user_filter(4gx)` — Service plugin for application of user login limits

1.10 Auxiliary services

- `pam_gromox(8gx)` — a PAM plugin to authenticate with Gromox
- `rtf2html(1gx)` — C++ helper for `php_mapi`'s `mapi_decompressrtf` function
- `adaptor(8gx)` — SQL data cache generator
- `event(8gx)` — Folder change notification daemon
- `midb(8gx)` — Message Index database daemon

1.11 System administration

- `gromox-abktconv(8gx)` — Utility for converting between ABKT and JSON
- `gromox-abktpull(8gx)` — Utility to extract ABKT templates from LDIF
- `gromox-dbop(8gx)` — Database maintenance utility
- `gromox-kdb2mt(8gx)` — Utility for analysis of Kopano databases
- `gromox-mailq(8gx)` — SMTP queue lister
- `gromox-mkmidb(8gx)` — Tool for creating a blank message index database
- `gromox-mkprivate(8gx)` — Tool for creating a blank private store
- `gromox-mkpublic(8gx)` — Tool for creating a blank public store
- `gromox-mt2exm(8gx)` — Utility for bulk-importing mail items into a Gromox store
- `gromox-pff2mt(8gx)` — Utility for analysis of PFF/PST/OST files
- `gromox-rebuild(8gx)` — Tool for rebuilding a message store
- `sa.cfg(5gx)` — configuration file for system administration CGI programs

1.12 Service plugins

- `at_client(4gx)` — Service plugin for deferred command execution with `at(1)`
- `authmgr(4gx)` — Demultiplexer for authentication requests
- `ip6_container(4gx)` — trivial source connection counter
- `ldap_adaptor(4gx)` — LDAP connector for authentication
- `logthru(4gx)` — service plugin for a stdout/file logger
- `mysql_adaptor(4gx)` — MySQL/MariaDB connector for user metadata and authentication
- `textmaps(4gx)` — Service plugin for various data maps
- `timer_agent(4gx)` — Service plugin for deferred command execution with `timer(8gx)`

1.13 Past commands that have been removed

The following is a list of programs that no longer exist. It is intended solely to capture keyword searches within the documentation for said obsolete/removed commands.

- `gromox-pffimport` — replaced by `gromox-pff2mt` and `gromox-mt2exm`.

1.14 Language bindings

- `mapi(4gx)` — PHP module providing MAPI functions

1.15 Listening sockets

- `/run/gromox/zcore.sock` — `zcore(8gx)`
- `*:24` — `delivery-queue(8gx)` LMTP/SMTP service (when Postfix is on 25)
- `:*25` — Normally, your own MTA (`postfix(1)`, `exim(8)`, whatever the case may be). `delivery-queue(8gx)` will only be on 25 in developer setups that wish to cut and skip Postfix/etc. to get a simpler test setup.
- `*:80` — `http(8gx)` HTTP service
- `*:110` — `pop3(8gx)` POP3 service
- `*:143` — `imap(8gx)` IMAP service
- `*:443` — `http(8gx)` HTTP over implicit TLS
- `*:993` — `imap(8gx)` IMAP over implicit TLS
- `*:995` — `pop3(8gx)` POP3 over implicit TLS
- `:::3344` — `zcore(8gx)` management console
- `:::4455` — `imap(8gx)` management console
- `:::5000` — `exmdb_provider(4gx)` plugin inside `http(8gx)`
- `:::5555` — `midb(8gx)` service
- `:::5566` — `delivery-queue(8gx)` management console
- `:::6666` — `timer(8gx)` service
- `:::7788` — `pop3(8gx)` management console

- [::1]:8899 — http(8gx) management console
- [::1]:9900 — midb(8gx) management console
- [::1]:22222 — pad(8gx) service
- [::1]:33333 — event(8gx) service

1.16 Files

The exact paths depend on the options used when Gromox's build was configured. Especially the path for libraries, represented in this documentation as `/usr/lib/gromox`, may for example actually be `/usr/lib64/gromox` or `/usr/lib/riscv64-linux-gnu`, depending on the platform.

- `/usr/lib/gromox/libgxf_*.so`: flusher plugins for delivery-queue(8gx)
- `/usr/lib/gromox/libgxx_*.so`: HTTP processing plugins for http(8gx)
- `/usr/lib/gromox/libgxm_*.so`: hook plugins for delivery(8gx)
- `/usr/lib/gromox/libgxp_*.so`: PDU processing plugins for http(8gx)
- `/usr/lib/gromox/libgxs_*.so`: service plugins
- `/var/lib/gromox/user`: directory hierarchy for private mailboxes
- `/var/lib/gromox/domain`: directory hierarchy for public mailboxes (public folders)
It is presently not possible to use a single directory for both mailbox types, since `exmdb_list.txt` uses the infix to determine the mode of access.
- `.../user/account@domain`: individual mailbox container
The directory name has no required form. It is entirely dependent upon the `users.maildir` column in MySQL. Some implementations of user management use a 2-level hierarchy, e.g. `/user/1/0`.
- `.../a@d/exmdb/exchange.sqlite3`: mail store with almost everything (no mail bodies)
- `.../a@d/cid/`: attachments and message bodies (PR_BODY, PR_HTML, PR_RTF_COMPRESSED).
- `.../a@d/eml/mid_string`: RFC 5322 representation for a message.
`mid_string` has no required form. Typically, there is `timestamp.seqid.hostname` which represents EMLs captured by delivery(8gx) on ingestion, and `timestamp.seqid.midb` for EMLs generated by midb(8gx) out of MAPI messages.
- `.../a@d/ext/mid_string`: Digest for the RFC 5322 file.
This JSON-encoded file contains e.g. indexing information for individual MIME parts of the RFC 5322 representation. Generated by midb(8gx).

GROMMUNIO-ADMIN

2.1 Name

`grommunio-admin` — `grommunio` admin CLI

2.2 Synopsis

`grommunio-admin -h`
`grommunio-admin` *COMMAND* [-h | *ARGS...*]

2.3 Description

Command line interface of the `grommunio` Admin API.

The CLI is not intended to provide the full functionality of the REST interface, but rather a low level administrative tool.

As the CLI is still under development, usage in automated scripts is generally discouraged.

The `-h/-help` option is not documented separately for each subcommand, but is valid at any point and prints usage information for the current subcommand.

The CLI supports color output if the `termcolor` module is installed.

2.4 Commands

2.4.1 `config`

Configuration introspection. See `grommunio-admin-config`.

2.4.2 connect

Connect to remote CLI. See *grommunio-admin-connect*.

2.4.3 dbconf

Database-stored configuration management. See *grommunio-admin-dbconf*.

2.4.4 domain

Domain management. See *grommunio-admin-domain*.

2.4.5 fetchmail

Fetchmail management. See *grommunio-admin-fetchmail*.

2.4.6 fs

Filesystem operations. See *grommunio-admin-fs*.

2.4.7 ldap

LDAP configuration, diagnostics and synchronization. See *grommunio-admin-ldap*.

2.4.8 mconf

Managed configurations manipulation. See *grommunio-admin-mconf*.

2.4.9 mlist

Mailing/distribution list management. See *grommunio-admin-mlist*.

2.4.10 passwd

User password management. See *grommunio-admin-passwd*.

2.4.11 run

Run the REST API. See *grommunio-admin-run*.

2.4.12 service

Control external services interface. See *grommunio-admin-service*

2.4.13 shell

Start interactive shell. See *grommunio-admin-shell*.

2.4.14 taginfo

Print information about proptags. See *grommunio-admin-taginfo*.

2.4.15 user

User management. See *grommunio-admin-user*.

2.4.16 version

Show version information. See *grommunio-admin-version*.

2.5 See Also

grommunio-dbconf

GROMMUNIO-ADMIN-CONFIG

3.1 Name

`grommunio-admin config` — `grommunio-admin config` introspection

3.2 Synopsis

`grommunio-admin config` *check*
`grommunio-admin config` (*dump* | **get**) [*KEY*]
`grommunio-admin config` *trace* [-*s*] (*files* | **values**) [*KEY*]

3.3 Commands

3.3.1 check

Check the structural validity of the configuration.

Does currently not validate the semantic integrity, i.e. existence of referenced files, LDAP or database connectivity etc., although this functionality may be added in the future.

3.3.2 dump, get

Print the complete configuration.

As the `grommunio-admin` configuration can (and probably will) be distributed over multiple files, the `get` command provides an easy way to see the effective configuration.

The output can be reduced to a single *KEY*, if specified. Sub-levels can be specified in dotted notation (e.g. `sync.defaultPolicy`)

The `dump` command is an alias for `get` and remains for backward compatibility.

3.3.3 trace

Trace source of effective configuration.

Results can be presented either by file (`files`), showing which parts of a file are actually used, or by value (`values`), showing which file each value originates from.

Installation of the Python `termcolor` package is advised for a more readable output. See section *Tracing* for more information.

3.4 Options

KEY Only view specified key.

-s, --show-history Display more value history (see section *Tracing* for more information)

3.5 Tracing

3.5.1 By-File

Print annotated contents of each file.

Each line is marked and color coded to show its status. The following annotations are used:

- `+`, green: The value is part of the final configuration
- `x`, red: The value is overwritten by a later file
- `*`, yellow: The object or list is extended by a later file
- `~`, grey: The value is overwritten with the same value

Additionally, lines overwriting or extending previous entries are printed in boldface.

When specifying `-show-history`, each value that is overwritten or extended is annotated with the files doing so (each being color coded with the effect it has on the value).

3.5.2 By-Value

Print annotated effective configuration.

Each line is annotated with the file it originates from. In case of objects and lists, all contributing files are listed.

When specifying `-show-history`, overwritten files containing that value are listed as well. The effective source file is underlined.

For better visualization, color coding is done on a per-file basis: Each file is assigned an individual style which is used for its contributions. Objects and lists originating from multiple files are always shown in boldface white.

3.6 See Also

`grommunio-admin`, `grommunio-admin-dbconf`, `grommunio-admin-mconf`

GROMMUNIO-ADMIN-CONNECT

4.1 Name

`grommunio-admin connect` — Connect to remote CLI

4.2 Synopsis

```
grommunio-admin connect [-c COMMAND] [-no-verify] [-redirect-fs [-auto-save (local|remote|discard|print)]] [-v] HOST [USER [PASSWORD]]
```

4.3 Description

Connect to a remote server to invoke CLI commands on.

Requires a running admin API with active remote CLI and a user with `SystemAdminPermission`.

Note that the remote CLI currently uses a REST interface which does not provide a standard input, rendering commands that rely on user interaction useless.

4.4 Options

HOST Host to connect to, in the format `protocol://hostname:port`, where protocol is either http or https. If omitted, the protocol is auto-detected, with https taking precedence over http. If no port is specified, the default ports 8080 (http) and 8443 (https) are used.

PASSWORD Password to use for authentication. Default is to prompt.

USER User to use for authentication. Defaults is `admin`.

--auto-save ACTION Choose automatic action for received files when filesystem redirection is enabled. Possible actions are:

discard - discard any received file

local - save at local path

print - print file contents to stdout and discard

remote - save at path reported from remote server

-c, --command Execute command on remote server and exit instead of starting an interactive shell.

- no-verify** Continue with https even if the TLS certificate presented by the server is invalid. Required if the server uses a self-signed certificate that is not installed on the system. Use with caution.
- redirect-fs** Redirect CLI initiated file operations to local filesystem. See section *Filesystem Emulation* for details.
- v, --verbose** Print more detailed information about the connection process.

4.5 Filesystem Emulation

When the *--redirect-fs* option is given, CLI initiated file operations are performed in an emulated filesystem and written files are sent back to the client.

Note that this does only apply to files which are opened by CLI operations, while module-level operations (e.g. loading of configurations) are unaffected.

Files received from the remote server can then be viewed or saved locally.

4.6 See Also

grommunio-admin, grommunio-admin-shell

GROMMUNIO-ADMIN-DBCONF

5.1 Name

`grommunio-admin dbconf` — Database-stored configuration management.

5.2 Synopsis

```
grommunio-admin dbconf (commit | delete) SERVICE [FILE [KEY]]  
grommunio-admin dbconf get SERVICE FILE [KEY]  
grommunio-admin dbconf list [SERVICE [FILE [KEY]]]  
grommunio-admin dbconf set [-b] [-i] [-] SERVICE FILE KEY VALUE
```

5.3 Description

grommunio dbconf provides the ability to store and manage configurations at a single location while making it available across distributed systems. The configurations are stored in the central MySQL database and can be accessed via *grommunio-dbconf* and *grommunio-admin-dbconf*.

While both tools essentially provide the same functionality, *grommunio-dbconf* provides far better performance and is intended to be used for quickly accessing the configuration.

Configurations consist of key/value pairs organized in files, grouped by service. Each service can have an arbitrary number of configuration files, which in turn can contain an arbitrary number of unique keys.

5.4 Commands

commit Trigger commit hook for service, file or key

delete Delete service, file or key

get Get file or single key

list List available services, files or keys

set Set a configuration key

5.5 Options

SERVICE Name of the service to configure

FILE Name of the configuration file

KEY Name of the configuration key

VALUE Value to store in the key

-- Indicate that all options have been specified and only names follow

-b, --batch Do not auto-commit

-i, --init Only set if configuration key does not exist yet

5.6 Commit Hooks

When modifying values, potential consumers can be notified of this change via commit hooks, for example by restarting the service using the configuration. For security reasons only a few white-listed commands are available (see section *AVAILABLE COMMIT COMMANDS*).

Commit hooks can be defined on key, file or service level. *set* operations always trigger commits at key level, while the *commit* command can directly trigger key or service level hooks depending on whether a file or key is specified.

If no hook is defined for a specific trigger level, it automatically falls through to the next lower level, in the order *key* > *file* > *service*.

Commit hooks for a service can be defined by setting `commit_key`, `commit_file` and `commit_service` keys under `grommunio-dbconf/<service>` to a valid command (see below).

5.7 Available Commit Commands

The following commands are available:

5.7.1 Key

```
postconf -e $ENTRY
```

5.7.2 File

There are currently no file-level commands.

5.7.3 Service

```
systemctl reload $SERVICE  
systemctl restart $SERVICE
```

5.8 Command Variable Expansion

Commands can contain `$`-prefixed variables that are expanded before execution. The literal `$$` can be used to generate a single `$`.

The following variables are valid:

ENTRY Expands to `$KEY=$VALUE` (key level only)

FILE Complete content of the modified file as newline separated key=value entries (file level only)

FILENAME Name of the modified file (key and file level)

KEY The modified key (key level only)

SERVICE Name of the modified service

VALUE New value of the modified key (key level only)

5.9 See Also

`grommunio-admin`, `grommunio-admin-config`, `grommunio-admin-mconf`, `grommunio-dbconf`

GROMMUNIO-ADMIN-DOMAIN

6.1 Name

grommunio-admin domain — Domain management

6.2 Synopsis

grommunio-admin domain create [*<FIELDS>*] [*-create-role*] [*-skip-adaptor-reload*] *-u MAXUSER*
DOMAINNAME

grommunio-admin domain delete *DOMAINSPEC*

grommunio-admin domain list [*-f FIELD=<value>*] [*-s FIELD*] [*DOMAINSPEC*]

grommunio-admin domain modify [*<FIELDS>*] *DOMAINSPEC*

grommunio-admin domain purge [*-files*] [*-y*] *DOMAINSPEC*

grommunio-admin domain recover *DOMAINSPEC*

grommunio-admin domain show [*-f FIELD=<value>*] [*-s FIELD*] *DOMAINSPEC*

6.3 Description

Subcommand to show and manipulate domains.

6.4 Commands

create Create a new domain

delete Soft-delete a domain

list List domains

modify Modify domain

purge Permanently delete domain

recover Recover a soft-deleted domain

show Show detailed information about a domain

6.5 Options

DOMAINNAME Complete name of the domain

DOMAINSPEC Domain name prefix or domain ID

--create-role Automatically create a domain administrator role for the new domain

--files Also delete files from disk

-f FIELD=<value>, **--filter FIELD=<value>** Filter expression in the form of 'field=value'. Can be specified multiple times to refine filter

--skip-adaptor-reload Do not reload gromox-adaptor service after domain creation

-s FIELD, **--sort FIELD** Sort by field. Can be given multiple times

-y, **--yes** Assume yes instead of prompting

6.6 Fields

--address ADDRESS Content of address field

--adminName ADMINNAME Name of the domain administrator or primary contact

--endDay ENDDAY Date of domain expiration in YYYY-MM-DD format

--orgID ID ID of the organization to assign the domain to

--tel TEL Telephone number of domain administrator or primary contact

-u MAXUSER, **--maxUser MAXUSER** Maximum number of users in the domain

6.7 See Also

grommunio-admin, **grommunio-admin-fs**, **grommunio-admin-user**

GROMMUNIO-ADMIN-FETCHMAIL

7.1 Name

grommunio-admin fetchmail — Manage fetchmail settings and generate rc file

7.2 Synopsis

```
grommunio-admin fetchmail create [<FIELDS>] -srcPassword PASSWORD -srcServer SERVER  
-srcUser USER USERSPEC [MAILBOX]  
grommunio-admin fetchmail delete [-y] MBSPEC  
grommunio-admin fetchmail list [-f FILTER] [-s SORT] [MBSPEC]  
grommunio-admin fetchmail modify [<FIELDS>] MBSPEC  
grommunio-admin fetchmail print [-q] MBSPEC  
grommunio-admin fetchmail show [-password] MBSPEC  
grommunio-admin fetchmail write-rc [-force] [-o FILE] [-p] [-t MINUTES] [-v]
```

7.3 Description

Subcommand to show and manipulate fetchmail entries and generate fetchmailrc file.

7.4 Commands

create Create a new fetchmail entry
delete Delete fetchmail entry
list List fetchmail entries
modify Modify fetchmail entry
print Print fetchmail configuration line generated by the entry
show Show detailed information about fetchmail entry
write-rc Write fetchmail configuration file (fetchmailrc)

7.5 Options

MAILBOX E-Mail address of the local mailbox to deliver the mails to. Defaults to e-mail address of the specified user

MBSPEC Mailbox prefix or ID of the fetchmail entry

USERSPEC Username prefix or ID of the user to attach the entry to

-f FIELD=<value>, **--filter FIELD=<value>** Filter expression in the form of 'field=value'. Can be specified multiple times to refine filter

--force Write rc file even if no entries were changed since the last write

-o, **--out-file** Path to write configuration to. Default is */etc/fetchmailrc*

--password Print the source password

-p, **--print** Additionally print rc file to stdout

-q, **--quiet** Do not print additional info

-s FIELD, **--sort FIELD** Sort by field. Can be given multiple times

-t, **--time** Time in minutes since the last write. Default is to autodetect by file mtime

-v, **--verbose** Be more verbose

-y, **--yes** Delete mutiple entries without prompting

7.6 Fields

--active STATE Whether the entry is active. STATE can be one of *0*, *1*, *yes* or *no*. Default is 1

--extraOptions EXTRAOPTIONS Space separated list of options to write into the fetchmailrc

--fetchall STATE Whether to fetch mails marked as seen on the source server. STATE can be one of *0*, *1*, *yes* or *no*. Default is 0

--keep STATE Whether to keep fetched mails on the source server. STATE can be one of *0*, *1*, *yes* or *no*. Default is 1

--protocol PROTOCOL Protocol to use for fetching. Can be one of *POP3*, *IMAP*, *POP2*, *ETRN* or *AUTO*. Default is IMAP

--srcAuth AUTH Authentication method to use. Can be one of *password*, *kerberos_v5*, *kerberos*, *kerberos_v4*, *gssapi*, *cram-md5*, *otp*, *ntlm*, *msn*, *ssh*, *any*. Default is password

--srcFolder FOLDER Source folder to fetch from

--srcPassword PASSWORD Password of the source user

--srcServer SERVER Source server to fetch from

--srcUser USER Source user to fetch mails from

--sslCertCheck STATE Whether to force SSL certificate check. STATE can be one of *0*, *1*, *yes* or *no*. Default is 0

--sslCertPath SSLCERTPATH Path to a directory containing trusted certificates or empty to use system default

--useSSL STATE Enable SSL

7.7 See Also

`grommunio-admin`, `grommunio-admin-user`

GROMMUNIO-ADMIN-FS

8.1 Name

grommunio-admin fs — Filesystem operations

8.2 Synopsis

```
grommunio-admin fs clean [-d] [-s] [PARTITION]  
grommunio-admin fs du [PARTITION]
```

8.3 Description

Show space used by user and domain home directories or remove unused files.
Unused files may remain when users or domains are deleted without removing their files.

8.4 Commands

clean Remove directories and files that are not used by any domain or user.
du Show data usage statistics

8.5 Options

PARTITION Apply only to selected partition. Can be either *domain* or *user*
-d, --dryrun Do not delete anything, just print what would be deleted
-s, --nostat Do not collect disk usage statistics of deleted files

8.6 See Also

`grommunio-admin`, `grommunio-admin-domain`, `grommunio-admin-user`

GROMMUNIO-ADMIN-LDAP

9.1 Name

grommunio-admin ldap — LDAP tools

9.2 Synopsis

```
grommunio-admin ldap check [-r [-m] [-y]]  
grommunio-admin ldap configure  
grommunio-admin ldap downsync [-a] [-c] [-f] [-y] [USER [USER ...]]  
grommunio-admin ldap dump USER  
grommunio-admin ldap info  
grommunio-admin ldap reload  
grommunio-admin ldap search [-n MAX_RESULTS] [USER]
```

9.3 Description

The grommunio admin ldap module provides functions for configuring and testing the LDAP connection and downloading or updating users.

9.4 Commands

check Check if the LDAP objects imported users are linked to can still be found, optionally removing orphaned users

configure Interactively configure or modify LDAP connection

downsync Synchronize or import users from LDAP

dump Print LDAP object

info Show connection status

reload Reload the LDAP configuration and reconnect

search Search for users

9.5 Options

USER LDAP object ID or search string

- a, --all** Synchronize all imported users. No new users are created, to import users use **-c**
- c, --complete** Import or update all users from the LDAP tree
- f, --force** Force update users that are linked to a different or no LDAP object
- m, --remove-maildirs** Also remove user files from disk
- n, --max-results** Maximum number of results or 0 to diable limit (default 25)
- r, --remove** Remove imported users of which the linked LDAP object could not be found
- y, --yes** Do not prompt, assume yes

9.6 See Also

grommunio-admin, grommunio-admin-fs, grommunio-admin-user

GROMMUNIO-ADMIN-MCONF

10.1 Name

`grommunio-admin mconf` — Managed configuration manipulation

10.2 Synopsis

```
grommunio-admin mconf dump CONFIG  
grommunio-admin mconf modify CONFIG unset KEY  
grommunio-admin mconf modify CONFIG ACTION [-i | -b] KEY VALUE  
grommunio-admin mconf print CONFIG  
grommunio-admin mconf reload CONFIG  
grommunio-admin mconf save CONFIG
```

10.3 Description

`grommunio` managed configuration (`mconf`) offers the possibility to manipulate configuration files used by `gromox`.

10.4 Commands

dump Print configuration file that would be generated from internal state
modify Modify internal configuration state
print Print internal configuration state
reload Reload configuration from disk
save Save configuration file to disk

10.5 Options

ACTION Modification action:

add - Add entry to list

remove - Remove entry from list

set - Add key

unset - Remove key

CONFIG Configuration file, either *authmgr* or *ldap*

KEY Configuration key

VALUE Configuration value for numeric or boolean values use *-b* and *-i* respectively

-b, --bool Convert value to boolean, valid values are *y, n, yes, no, true, false, 1, 0*

-i, --int Convert value to integer, octal (*0o*) and hexadecimal (*0x*) prefixes are supported

10.6 See Also

grommunio-admin, **grommunio-admin-config**, **grommunio-admin-dbconf****, ****grommunio-admin-ldap****

GROMMUNIO-ADMIN-MLIST

11.1 Name

grommunio-admin mlist — Mailing/distribution list management

11.2 Synopsis

```
grommunio-admin mlist add MLISTSPEC (sender|recipient) ENTRY grommunio-admin mlist create
[-c CLASS] [-p PRIVILEGE] [-r RECIPIENT] [-s SENDER] [-t TYPE] NAME
grommunio-admin mlist delete [-y] MLISTSPEC
grommunio-admin mlist list [-f FIELD= <value>] [-s FIELD] [MLISTSPEC]
grommunio-admin mlist modify [-c CLASS] [-p PRIVILEGE] [-r RECIPIENT] MLISTSPEC
grommunio-admin mlist remove MLISTSPEC (sender|recipient) ENTRY grommunio-admin mlist
show
```

11.3 Description

Create, modify or delete mailing lists.

11.4 Commands

add Add sender or recipient to list
create Create a new mailing list
delete Delete mailing list
list List mailing lists
modify Modify mailing list
remove Remove sender or recipient from list
show Show detailed information about mailing list

11.5 Options

- c CLASS, --class CLASS ID of the associated class (class type only)
- p PRIVILEGE, --privilege PRIVILEGE Set who is allowed to send mails to the list, one of *all*, *domain*, *internal*, *outgoing* or *specific*
- f FIELD=<value>, --filter FIELD=<value> Filter expression in the form of 'field=value'. Can be specified multiple times to refine filter
- s FIELD, --sort FIELD Sort by field. Can be given multiple times
- t TYPE, --type TYPE List type (recipient selection), one of *normal*, *domain* or *class*

11.6 See Also

grommunio-admin, **grommunio-admin-domain**, **grommunio-admin-user**

GROMMUNIO-ADMIN-PASSWD

12.1 Name

`grommunio-admin passwd` — Set user password

12.2 Synopsis

`grommunio-admin passwd` [-a] [-l *LENGTH*] [-p *PASSWORD*] [*USER*]

12.3 Description

Set user password.

If no user is specified, the password is set for the *admin* user, which is created automatically if necessary.

If neither *-a* nor *-p* is provided, the user is prompted for a password.

12.4 Options

USER User to set password for (default *admin*)

-a, --auto Automatically generate a password

-l LENGTH, --length LENGTH Length of the automatically generated password (default 16)

-p PASSWORD, --password PASSWORD Password to set (do not prompt)

12.5 See Also

`grommunio-admin`, `grommunio-admin-user`

GROMMUNIO-ADMIN-RUN

13.1 Name

`grommunio-admin run` — Start a stand-alone HTTP server

13.2 Synopsis

`grommunio-admin run` [-d] [-i IP] [-no-config-check] [-p PORT]

13.3 Description

Run REST API in a stand-alone HTTP server.

—DO NOT USE IN PRODUCTION!—

This command is intended for development and testing. A production instance should use an external WSGI server like `uwsgi`.

13.4 Options

- d, --debug Enable debug mode
- i IP, --ip IP Host address to bind to (default 0.0.0.0)
- no-config-check Skip configuration check
- p PORT, --port PORT Host port to bind to (default 5001)

13.5 See Also

`grommunio-admin`, `grommunio-admin-config`

GROMMUNIO-ADMIN-SERVICE

14.1 Name

grommunio-admin service — grommunio-admin external service interface control

14.2 Synopsis

```
grommunio-admin service disable [SERVICE [SERVICE [. . . ]]]  
grommunio-admin service [-r] load [SERVICE [SERVICE [. . . ]]]  
grommunio-admin service [-v] status
```

14.3 Description

grommunio-admin connects to several external services to either provide means of configuration via API (e.g. grommunio chat) or to retrieve additional information (e.g. LDAP).

grommunio-admin service can be used to introspect the connection status of these services.

Note that the CLI runs separately from the API backend. If introspection of the running server instance is required, use the *connect* command to access the server instance.

Each service has a state, reflecting the connection status. The following states are used:

UNLOADED The service has not been loaded yet. It will be loaded automatically when needed.

LOADED The service has been initialized successfully.

UNAVAILABLE An error occurred that indicates that the service is not available, but might become available in the future. No reload is necessary to reconnect.

SUSPENDED An error occurred that indicates that the service is not available, but might become available in the future. The service will be reloaded automatically on next usage.

ERROR The service is not available either because initialization failed or because to man errors occurred. It will remain unavailable until reloaded manually.

DISABLED The service has been manually disabled (either by configuration or command).

14.4 Commands

14.4.1 disable

Disable the service. Requires reload to enable.

14.4.2 load

Load or reload services.

Only services in UNLOADED or SUSPENDED state will be affected unless the *-reload* option is given.

14.4.3 status

Show status of all services.

14.5 Options

SERVICE Name of the service.

-r, --reload Force reload of service.

-v, --verbose Show more information.

14.6 Services

The following services are currently connected via the service interface:

chat grommunio chat. Connected via REST interface.

exmdb gromox exmdb provider (gromox-http). Connected via custom TCP protocol.

ldap External LDAP service. Connected via LDAP(s).

redis Redis instance (used by grommunio sync). Connected via redis driver (TCP).

systemd Systemd shell execution.

14.7 See Also

grommunio-admin, **grommunio-admin-config**, **grommunio-admin-connect**, **grommunio-admin-mconf**

GROMMUNIO-ADMIN-SHELL

15.1 Name

`grommunio-admin shell` — Start interactive shell

15.2 Synopsis

`grommunio-admin shell` [-d] [-n] [-x]

15.3 Description

The interactive shell mode allows execution of multiple (new line separated) commands in a single session. Command syntax is identical to the CLI arguments, with addition of the *exit* command which ends the interactive shell.

If possible, typed history will be saved in `~/.grommunio-admin.history`.

15.4 Options

`-d, --debug` Enable more verbose debug output

`-n, --no-history` Disable loading/saving of the typed history

`-x, --exit` Exit immediately if a command results in a non-zero exit code

15.5 See Also

`grommunio-admin`, `grommunio-admin-connect`

GROMMUNIO-ADMIN-TAGINFO

16.1 Name

`grommunio-admin taginfo` — Show information about proptags

16.2 Synopsis

`grommunio-admin taginfo TAG [TAG ...]`

16.3 Description

Display information about a property tag, as defined in the Microsoft Exchange Server Protocols Master Property List.

Note that property names used by grommunio may differ from the names defined by Microsoft.

16.4 Options

TAG Decimal or hexadecimal (with 0x prefix) Tag ID or grommunio tag name

16.5 See Also

`grommunio-admin`

GROMMUNIO-ADMIN-USER

17.1 Name

grommunio-admin user — User management

17.2 Synopsis

```
grommunio-admin user delete [-k] [-y] USERSPEC  
grommunio-admin user list [-f FIELD=<value>] [-s FIELD] [USERSPEC]  
grommunio-admin user show [-f FIELD=<value>] [-s FIELD] USERSPEC
```

17.3 Description

Subcommand to show and delete users.

No functionality for adding or modifying users is implemented at the moment.

17.4 Commands

delete Delete user

list List users

show Show detailed information about a user

17.5 Options

USERSPEC User name prefix or user ID

-f FIELD=<value>, **--filter FIELD=<value>** Filter expression in the form of 'field=value'. Can be specified multiple times to refine filter

-k, **--keep-files** Do not delete user files from disk

-s FIELD, **--sort FIELD** Sort by field. Can be given multiple times

-y, **--yes** Assume yes instead of prompting

17.6 See Also

`grommunio-admin`, `grommunio-admin-domain`, `grommunio-admin-fs`, `grommunio-admin-ldap`,
`grommunio-admin-passwd`

GROMMUNIO-ADMIN-VERSION

18.1 Name

`grommunio-admin version` — Show backend and/or API version

18.2 Synopsis

`grommunio-admin version [-a] [-b] [-c]`

18.3 Description

Show the current version of the API (specification) or the backend (code).

The combined mode (default) appends the difference between between backend and API version at the end of the API version.

If multiple options are given, each requested version is printed on a separate line. The order is always API – backend – combined.

18.4 Options

`-a, --api` Print API version

`-b, --backend` Print backend version

`-c, --combined` Print combined version

18.5 See Also

`grommunio-admin`

GROMMUNIO-DBCONF

19.1 Name

`grommunio-dbconf` — Database-stored configuration tool

19.2 Synopsis

grommunio-dbconf [-v] [-] (*commit* | *delete*) *SERVICE* [*FILE* [*KEY*]]

grommunio-dbconf *get* [-v] [-] *SERVICE FILE* [*KEY*]

grommunio-dbconf *list* [-v] [-] [*SERVICE* [*FILE* [*KEY*]]]

grommunio-dbconf *set* [-b] [-i] [-v] [-] *SERVICE FILE KEY* [*VALUE*]

19.3 Description

grommunio dbconf provides the ability to store and manage configurations at a single location while making it available across distributed systems. The configurations are stored in the central MySQL database and can be accessed via *grommunio-dbconf* and *grommunio-admin-dbconf*.

While both tools essentially provide the same functionality, *grommunio-dbconf* provides far better performance and is intended to be used for quickly accessing the configuration.

Configurations consist of key/value pairs organized in files, grouped by service. Each service can have an arbitrary number of configuration files, which in turn can contain an arbitrary number of unique keys.

19.4 Commands

commit Trigger commit hook for service, file or key

delete Delete service, file or key

get Get file or single key

list List available services, files or keys

set Set a configuration key

19.5 Options

SERVICE Name of the service to configure

FILE Name of the configuration file

KEY Name of the configuration key

VALUE Value to store in the key

-- Indicate that all options have been specified and only names follow

-b, --batch Do not auto-commit

-i, --init Only set if configuration key does not exist yet

-v, --verbose Increase verbosity level (up to 3) to produce more diagnostic output

19.6 Commit Hooks

When modifying values, potential consumers can be notified of this change via commit hooks, for example by restarting the service using the configuration. For security reasons only a few white-listed commands are available (see section **AVAILABLE COMMIT COMMANDS**).

Commit hooks can be defined on key, file or service level. *set* operations always trigger commits at key level, while the *commit* command can directly trigger key or service level hooks depending on whether a file or key is specified.

If no hook is defined for a specific trigger level, it automatically falls through to the next lower level, in the order *key* > *file* > *service*.

Commit hooks for a service can be defined by setting `commit_key`, `commit_file` and `commit_service` keys under `grommunio-dbconf/<service>` to a valid command (see below).

19.7 Available Commit Commands

The following commands are available:

19.7.1 Key

```
postconf -e $ENTRY
```

19.7.2 File

There are currently no file-level commands.

19.7.3 Service

```
systemctl reload $SERVICE
systemctl restart $SERVICE
```

19.8 Command Variable Expansion

Commands can contain `$`-prefixed variables that are expanded before execution. The literal `$$` can be used to generate a single `$`.

The following variables are valid:

ENTRY Expands to `$KEY=$VALUE` (key level only)

FILE Complete content of the modified file as newline separated key=value entries (file level only)

FILENAME Name of the modified file (key and file level)

KEY The modified key (key level only)

SERVICE Name of the modified service

VALUE New value of the modified key (key level only)

19.9 See Also

grommunio-admin, **grommunio-admin-mconf**, **grommunio-dbconf**

20.1 Name

adaptor — SQL data cache generator

20.2 Synopsis

adaptor [-c *config*]

20.3 Description

adaptor reads table data from MySQL and builds text files that act as caches for modules that utilize them. See the FILES section for example uses.

20.4 Options

- c *config* Read configuration directives from the given file. If this option is not specified, /etc/gromox/adaptor.cfg will be read if it exists.
- version Output version information and exit.
- ? Display option summary.

20.5 Files

- *state_path*/alias_addresses.txt: generated by adaptor from MySQL data, used by alias_translator(4gx).
- *state_path*/console_table.txt: static file (edited by admin only) telling adaptor which consoles to contact to issue reload commands.

20.6 Configuration file directives

collector_mailbox Default: (unset)

limit_type Default: (unset)

log_file_path Default: */var/log/gromox/sa.log*

mysql_dbname Default: *email*

mysql_host

The hostname/IP address for contacting the SQL server.

Default: *localhost*

mysql_password Default: (unset)

mysql_port

The TCP port number for contacting the SQL server.

Default: *3306*

mysql_username Default: *root*

state_path

Directory for additional files.

Default: */var/lib/gromox*

subsystem_address Default: (unset)

20.7 See also

gromox(7)

ALIAS_TRANSLATOR

21.1 Name

`alias_translator` — Alias resolution plugin for `delivery(8gx)`

21.2 Description

`alias_translator` is a mail transfer agent hook plugin which rewrites the envelope FROM and RCPT addresses of mails and replaces aliases by their respective primary addresses.

Alias mappings are read from `state_path/alias_addresses.txt`, whereby `state_path` is the eponymous directive from the program that loaded the plugin. The `alias_address.txt` file constitutes a caching mechanism and is normally generated from MySQL data by `adaptor(8gx)`. The adaptor process is also the one triggering the reload in `alias_translator(4gx)` via `delivery(8gx)`'s telnet console mechanism.

21.3 See also

`gromox(7)`, `adaptor(8gx)`, `delivery(8gx)`

22.1 Name

`at_client` — Service plugin for deferred command execution with `at(1)`

22.2 Description

`at_client` is a loadable plugin that will utilize the `at(1)` framework to schedule deferred commands.

In practice, this is used by `exchange_emsmb(4gx)` and `zcore(8gx)` to implement delayed sending of messages.

The `atd(8)` daemon needs to be running for commands to get executed.

22.3 Configuration file directives

This plugin has no directives.

22.4 Caveats

The `at` implementation of Thomas Koenig and maintained by Calhariz (as of 2020) is present in Linux distributions as well as FreeBSD, and it wraps the job ID counter at 1048576 back to 0. New jobs can receive an ID that is still in use by a prior pending job.

The `at` implementation of OpenBSD (derived from Thomas Koenig's) has a modification by Millert that changes job IDs into job “names”, which are of the form “<starttime>.<queue>”, i.e. no longer a single integer. This is unsupported by Gromox. Due to the job names, the `at` implementation supports storing at most one job every second — the start time is shifted forwards if a collision is detected, but only up to 120 seconds, which means that job enqueueing can trivially fail.

22.5 See also

`gromox(7)`, `at(1)`, `atd(8)`

23.1 Name

authmgr(4gx) — Demultiplexer for authentication requests

23.2 Description

authmgr is a service plugin that, for mail account authentication, dynamically selects the authentication backend per user account. The `libgxs_mysql_adaptor.so` and/or `libgxs_ldap_adaptor.so` plugins need to be loaded in advance (this is the case by default).

23.3 Configuration file directives

auth_backend_selection

This controls how authmgr will verify passwords supplied with login operations. See the “Authentication modes” section below for details.

Available: *deny_all*, *allow_all*, *always_mysql*, *always_ldap*, *externid*

Default: *externid*

23.4 Authentication modes

- *deny_all* rejects every attempt at authentication. This is at best useful for testing.
- *allow_all* permits every attempt at authentication (provided the user exists). This may be handy when doing the initial mass-import of mailboxes via external IMAP-to-IMAP synchronization utilities such as `imapsync` without needing to know user passwords.
- *always_mysql* will cause authmgr to exclusively attempt password verification with the password field in the MySQL database.
- *always_ldap* will cause authmgr to exclusively attempt password verification with LDAP servers.
- *externid* will cause authmgr to selectively pick LDAP/MySQL, depending on whether the user was imported from LDAP or not.

23.5 See also

`gromox(7)`, `ldap_adaptor(4gx)`, `mysql_adaptor(4gx)`

AUTODISCOVER

24.1 Name

Autodiscover HTTP Service Protocol handler

24.2 Description

Autodiscover clients can locate the Autodiscover server for a given domain *example.com* by attempting to resolve the **_autodiscover._tcp.example.com** IN SRV record from DNS, and otherwise fall back to **autodiscover.example.com**.

To force using a particular Autodiscover server, such as when Gromox is run in a development environment with a fake domain, `c:windowssystem32driversetchosts` can be populated with a static entry for **autodiscover.example.com** to get that particular scenario working.

An Autodiscover client would make a “POST /Autodiscover/Autodiscover.xml” request to the autodiscover server that was determined moments ago. The mailbox credentials are presented using HTTP authentication.

In a Gromox minimal setup, `http(8gx)` will receive this POST request and, in conjunction with the built-in defaults of `mod_rewrite(4gx)` and `mod_fastcgi(4gx)`, is forwarded to a `php-fpm` instance, since the Autodiscover handler is coded in PHP. Alternative setups where, for example, an `nginx` frontend is used, `nginx` can also be the server that passes the request to a PHP executor.

The Autodiscover response contains a HTTP server (generally the HTTP home server) and the designated choice for protocol framing. A client uses this to set up the EMSMDB MAPI service within a MAPI profile. Because the HTTP home server is then known, Autodiscover is not used again when making a connection to the message store service. However, the Address Book always issues Autodiscover requests. (In other words, removing the DNS entry for the Autodiscover server after a profile is set up would break the address book, but not the message store.)

24.3 Configuration file directives

The configuration file is `/etc/gromox/autodiscover.ini`. In its absence, built-in defaults, tuned for the minimal Gromox installation, are used. A few ini sections and keys are recognized.

24.3.1 [database]

host Host and optionally port to pass to php-mysqli. Default: *localhost*.

username User for the SQL connection. Default: *root*.

password Password for the same. Default: (empty string).

dbname Database name. Default: *email*.

24.3.2 [exchange]

hostname The hostname of the Autodiscover server. This is used by the Autodiscovery handler to issue HTTP->HTTPS upgrades. Default: (system hostname).

mapihhttp

This setting controls which server types the Autodiscover handler should respond with. The two options are MAPI-in-DCERPC-in-HTTP (a.k.a. RPCH, RPC/HTTP, Outlook Anywhere) and simply MAPI-in-HTTP (a.k.a. MOH, MAPI-over-HTTP). MOH is supported from Outlook >= 2013 SP1 onwards. *0* selects RPC/HTTP, *1* selects MAPI/HTTP.

Default: *0*.

24.3.3 [default]

timezone Default: (unspecified)

24.3.4 [system]

24.3.5 [http-proxy]

This section contains the emsmdb multiserver map. This conveys the HTTP home server for users. If a HTTP home server is not also the exmdb/sqlite home server for a user, said HTTP server will proxy the datastream to the right exmdb home server, which may result in lower performance than when connecting to the right server from the start.

Keys in this ini section are homedir prefixes, and values are the corresponding HTTP server, quite similar to exmdb_list.txt.

Default:

```
/var/lib/gromox/user/ = (system hostname)
/var/lib/gromox/domain/ = (system hostname)
```

24.4 Normative references

- MS-OXDISCO: Autodiscover HTTP Service Protocol
- MS-OXDSCCLI: Autodiscover Publishing and Lookup Protocol

24.5 See also

gromox(7)

DELIVERY-QUEUE

25.1 Name

delivery-queue — LMTP/SMTP frontend for local delivery

25.2 Synopsis

delivery-queue [-c *config*]

25.3 Description

delivery-queue is an additional mail queue in front of delivery(8gx). It may be removed in a future version.

25.4 Options

-c *config* Read configuration directives from the given file. If this option is not specified, /etc/gromox/smtp.cfg will be read if it exists.

-? Display option summary.

25.5 Configuration file directives

block_interval_sessions

The amount of time a source IP address is blocked from connecting to the service because it tried to queue too many messages.

Default: *1 minute*

command_protocol

Selects what kind of hello command is accepted of clients. Can be *lmtp*, *smtp* or *both*.

Default: *both*

config_file_path

Colon-separated list of directories which will be scanned when locating further configuration files, especially those used by plugin instances.

Default: */etc/gromox/smtp:/etc/gromox*

console_server_ip

An IPv6 address (or v4-mapped address) to expose the management console frontend on.

Default: `::1`

console_server_port

The TCP port number to expose the management console frontend on.

Default: `5566`

context_average_mem Default: `256K`

context_max_mem Default: `2M`

context_num Default: `200`

data_file_path

Colon-separated list of directories which will be scanned when locating data files.

Default: `/usr/share/gromox/smt`

default_domain Default: (system domainname)

domain_list_valid Default: *false* if server mode is 0 (incoming), *true* otherwise

flusher_plugin_path

Filename of a flusher plugin to load and use. (Only one can be in use at any time.)

Default: `libgxf_message_enqueue.so`

host_id Default: (system hostname)

listen_port

The TCP port to export the SMTP protocol service on.

Default: `25`

listen_ssl_port

The TCP port to expose the implicit-TLS SMTP protocol service on.

Default: (unset)

mail_max_length

Maximum permitted length of a message.

Default: `64M`

running_identity

An unprivileged user account to switch the process to after startup. To inhibit the switch, assign the empty value.

Default: `gromox`

service_plugin_ignore_errors

If set to yes, service plugins that fail to load on startup are ignored. If set to no, the daemon will exit if any plugin cannot be loaded.

Default: `no`

service_plugin_list

Path to a text file which lists the filenames of service plugins to load, one per line.

Default: (unspecified)

service_plugin_path

Path to a secondary directory where service plugins will be loaded from if a primary search in standard directories (as per `ld.so(8)`) was unsuccessful.

Default: `/usr/lib/gromox`

smtp_certificate_passwd

The password to unlock TLS certificates.

Default: (unset)

smtp_certificate_path

Filesystem path to a certificate file for use with encrypted connection. The complete certificate chain should be present (as there is no other config directive to pull CA certs in).

Default: (unset)

smtp_conn_timeout

If an SMTP connection stalls for the given period, the connection is terminated.

Default: *3 minutes*

smtp_force_starttls

This flag controls whether clients must utilize TLS, either by way of implicit TLS (cf. **listen_ssl_port**), or through the STARTTLS command.

Default: *false*

smtp_max_mail_num

The maximum number of messages that a client is allowed to queue within one connection before its source IP address is blocked.

Default: *10*

smtp_private_key_path

Filesystem path to the key file needed to unlock the TLS certificate.

Default: (unset)

smtp_support_pipeline

This flag controls the offering of the PIPELINING extension (RFC 2920) to clients.

Default: *true*

smtp_support_starttls

This flag controls the offering of the STARTTLS extension (RFC 3027) to clients.

Default: *false*

state_path

Directory for runtime variadic data.

Default: */var/lib/gromox*

thread_charge_num

The maximum number of connections that each thread is allowed to process.

Default: *40*

thread_init_num

The minimum number of client processing threads to keep around.

Default: *1*

25.6 Files

- *data_file_path*/smtp_code.txt: Mapping from internal SMTP error codes to textual descriptions.
- /usr/lib/gromox/libgxs_*.so: flusher plugins
- /usr/lib/gromox/libgxs_*.so: service plugins

25.7 See also

gromox(7), **delivery(8gx)**, **message_enqueue(4gx)**, **midb_agent(4gx)**

26.1 Name

delivery(8gx) — Backend for local delivery

26.2 Synopsis

delivery [-c *config*]

26.3 Options

-c *config* Read configuration directives from the given file. If this option is not specified, /etc/gromox/delivery.cfg will be read if it exists.

-? Display option summary.

26.4 Plugins

The delivery daemon can be extended by a number of plugins. In Gromox, their filenames begin with **libmtahook_**.

26.5 Configuration file directives

admin_mailbox Default: (unspecified)

config_file_path

Colon-separated list of directories in which further configuration files, especially those used by plugin instances, will be searched.

Default: /etc/gromox/delivery:/etc/gromox

console_server_ip

An IPv6 address (or v4-mapped address) to expose the management console frontend on.

Default: ::1

console_server_port

The TCP port number to expose the management console frontend on.

Default: 6677

data_file_path

Colon-separated list of directories in which static data files will be searched.

Default: */usr/share/gromox/delivery*

default_domain Default: (system domainname)

dequeue_max_mem Default: *1024M*

dequeue_path Default: */var/lib/gromox/queue*

domain_list_valid Default: *true*

free_context_num Default: *512*

host_id Default: (system hostname)

mpc_plugin_ignore_errors

If set to yes, MPC plugins that fail to load on startup are ignored. If set to no, the daemon will exit if any plugin cannot be loaded.

Default: *no*

mpc_plugin_list

Path to a text file which lists the filenames of MPC plugins to load, one per line.

Default: (unspecified)

mpc_plugin_path

Path to a secondary directory where MPC plugins will be loaded from if a primary search in standard directories (as per *ld.so(8)*) was unsuccessful.

Default: */usr/lib/gromox*

running_identity

An unprivileged user account to switch the process to after startup.

Default: *gromox*

service_plugin_ignore_errors

If set to yes, service plugins that fail to load on startup are ignored. If set to no, the daemon will exit if any plugin cannot be loaded.

Default: *no*

service_plugin_list

Path to a text file which lists the filenames of service plugins to load, one per line.

Default: (unspecified)

service_plugin_path

Path to a secondary directory where service plugins will be loaded from if a primary search in standard directories (as per *ld.so(8)*) was unsuccessful.

Default: */usr/lib/gromox*

state_path

Directory for runtime variadic data.

Default: */var/lib/gromox*

work_threads_max Default: *32*

work_threads_min Default: *16*

26.6 Files

- /usr/lib/gromox/libgxm_*.so: hook plugins
- /usr/lib/gromox/libgxs_*.so: service plugins

26.7 See also

gromox(7)

27.1 Name

`str_table` — string table match service plugin

27.2 Description

`str_table` implements checking for a trivial presence lookup of something in a text file. The plugin can be used in any Gromox process, but in practice only makes an appearance in `delivery-queue(8gx)` and `delivery(8gx)`.

The `str_table` plugin has multiple instances. These are: **`domain_list`**.

These instances are factual copies of the module to please the plugin loader and make it load different `.cfg` files. (This is clearly inefficient and may be fixed in a future release.) For example, the plugin loader will derive the name “`domain_list.cfg`” from the module name “`libgxs_domain_list.so`”, and then pass this filename to the module so it can be loaded as a config file.

The plugin supports temporary modification of the string table as it exists in memory through the telnet console mechanism. A reload of the table from disk purges these modifications.

27.3 Configuration file directives

`add_service_name`

The plugin instance will expose its function through this service function name. You should not generally this, because other plugins rely on certain fixed values.

Default: *instance_name_add*

`growing_num`

The maximum number of additional string table entries on top of what the on-disk table file has supplied. So when the table file contains 40 entries and `growing_num` is 100, the in-memory copy of the table can hold up to 140 entries total.

Default: *100*

`is_case_sensitive`

Flag which determines whether matches should treat uppercase and lowercase differently or not.

Default: *false*

`query_service_name`

The plugin instance will expose its function through this service function name. You should not generally this, because other plugins rely on certain fixed values.

Default: *instance_name_query*

remove_service_name

The plugin instance will expose its function through this service function name. You should not generally this, because other plugins rely on certain fixed values.

Default: *instance_name_remove*

27.4 Files

- *config_file_path/instance.cfg*: configuration file for the instance of *str_table* (e.g. */etc/gromox/delivery/domain_list.cfg* when *delivery(8gx)* was made to load *libgxs_domain_list.so*.)
- *state_path/instance.txt*, *config_file_path/instance.txt*: string table on which matches are carried out (e.g. */var/lib/gromox/domain_list.txt*)

27.5 See also

gromox(7)

28.1 Name

event — Folder change notification daemon

28.2 Synopsis

event [-c *config*]

28.3 Description

The event daemon is a software bus, inter-process communication (IPC) mechanism that allows communication between multiple processes running concurrently on multiple machines.

In practice, it is used by midb(8gx), pop3(8gx) and imap(8gx) to notify imap(8gx) instances of changed folder/message states.

28.4 Options

-c *config* Read configuration directives from the given file. If this option is not specified, `/etc/gromox/event.cfg` will be read if it exists.

-version Output version information and exit.

-? Display option summary.

28.5 Files

- `config_file_path/event_acl.txt`: A file with one address (IPv6 or v4-mapped) per line of allowed clients. In its absence, `::1` is default-whitelisted.

28.6 Configuration file directives

config_file_path

Colon-separated list of directories which will be scanned when locating further configuration files.
Default: `/etc/gromox/event:/etc/gromox`

event_listen_ip

An IPv6 address (or v4-mapped address) for exposing the event service on.
Default: `::1`

event_listen_port

The TCP port number for exposing the event service on.
Default: `33333`

event_threads_num

The minimum number of client processing threads to keep around.
Default: `50`

running_identity

An unprivileged user account to switch the process to after startup. To inhibit the switch, assign the empty value.
Default: `gromox`

28.7 Event protocol

The event service is exposed as a line-based text protocol. Upon connection, the event server gratuitously writes “OK” and will wait for commands. Each connection to the event daemon starts out in Enqueue Mode, and this is the only mode from which commands can be issued.

“FALSE” may be emitted by the server if there is a syntax error.

The command “ID <res_id>” declares the particular connection to be a notification sender. `res_id` is generally the hostname and the PID. The server always responds with “TRUE”. (The connection stays in Enqueue Mode.)

The command “LISTEN <res_id>” declares the particular connection to be a notification receiver. `res_id` follows the same pattern. The server responds with “TRUE” and the connection state changes to the Dequeue Mode (see below).

The command “SELECT <username> <folder>” subscribes those connections that have registered **as a listener for res_id** to notifications. (This means that a process wishing to use `event_stub(4gx)` to listen for notifications strictly requires loading `event_proxy(4gx)` too, and, in essence, use two connections to `event(8gx)`.) The server responds with “FALSE” if no listener exists, or “TRUE” on success.

The command “UNSELECT <username> <folder>” unsubscribes those connections that had registered as a listener for `res_id`. The server always responds with “TRUE”.

Auxiliary self-explanatory commands available are: “QUIT” and “PING”.

Any other input is treated as a notification item and is not interpreted by `event(8gx)` beyond checking the number of fields:

The notification “FOLDER-TOUCH <username> <folder>” informs listeners that the folder metadata has changed and warrants being reloaded.

The notification “MESSAGE-FLAG <username> <folder> <messageid>” informs listeners that the message metadata has changed and warrants being reloaded.

Clients in Dequeue Mode will receive notifications. Each notification line received by the client needs to be acknowledged with a “TRUE” response. It is not possible to exit Dequeue Mode; connection termination is the only way out.

28.8 See also

gromox(7), event_proxy(4gx), event_stub(4gx)

29.1 Name

event_stub — event sender service plugin

29.2 Description

event_proxy connects to the event(8gx) daemon and sets itself up so as to be able to send notifications.

event_proxy installs three service functions, “broadcast_event”, “broadcast_select” and “broadcast_unselect”, though the latter two are just convenience functions for broadcast_event. broadcast_event is for synchronously sending a notification into the event distribution system. Arbitrary notifications and commands can be sent this way. The return value is ignored.

In practice, midb(8gx), imap(8gx) and pop3(8gx) issue FOLDER-TOUCH notifications. Only imap(8gx) issues MESSAGE-FLAG notifications.

29.3 See also

event(8gx), event_stub(4gx)

EVENT_STUB

30.1 Name

event_stub — event receiver service plugin

30.2 Description

event_stub connects to the event(8gx) daemon and sets itself up to receive notifications asynchronously with the help of an extra thread. (No notifications are sent towards event(8gx) by this service plugin.)

event_stub installs one service function, “install_event_stub”, with which imap(8gx) registers a callback function that, in turn, is invoked whenever a notification is received from the event daemon.

In practice, imap handles FOLDER-TOUCH and MESSAGE-FLAG notifications received through this event channel.

30.3 See also

event(8gx), event_proxy(4gx)

EXCHANGE_EMSEMB

31.1 Name

http(8gx) processing plugin for the Wire Format Protocol (Outlook/Exchange RPCs).

31.2 Description

exchange_emsmdb is a processing plugin for http(8gx) which handles the remote procedure calls for the EMSMDB v1 and AsyncEMSMDB v1 RPC interfaces, as well as OXCROPS as issued by the EcDoRpcExt2 call or the MAPIHTTP EXECUTE call.

EMSMDB is a DCE/RPC interface with just a few RPC calls (6 are still used today). ecDoRpcExt2, a call offered by that interface, takes an opaque byte buffer argument not interpreted by DCE/RPC. That byte buffer contains another protocol, "Remote Operation(s) Encoding Protocol" [OXCROPS]. No reason for this wrapping is given in the OXCRPC spec. MAPIHTTP runs OXCROPS directly without the extra EMSMDB framing.

OXCROPS consists of 130 calls that make up the mailbox protocol.

31.3 Config file directives

async_threads_num Default: 4

average_handles Default: 1000

average_mem Default: 4K

mailbox_ping_interval Default: 5 minutes

mail_max_length Default: 64M

max_ext_rule_length Default: 510K

max_mail_num Default: 1000000

max_rcpt_num

The maximum number of recipients that an e-mail is allowed to have.

Default: 256

rop_debug

Log every incoming OXCROP call and the return code of the operation in a minimal fashion to stderr. Level 1 emits RPCs with a failure return code, level 2 emits all RPCs.

Default: 0

separator_for_bounce Default: ;

smtp_server_ip

SMTP server hostname or address to contact for outgoing mail.

Default: `::1`

smtp_server_port

SMTP server TCP port number to connect to for outgoing mail.

Default: `25`

submit_command Default: `/usr/bin/php /usr/share/gromox/sa/submit.php`

x500_org_name Default: (unspecified)

31.4 Files

- `data_file_path/notify_bounce/`: response templates for when mail cannot be delivered
- `data_file_path/msgchg_grouping/.txt*`

31.5 Normative references

- DCERPC / C706: Technical Standard DCE 1.1: Remote Procedure Call by The Open Group, 1997
- MS-OXCRPC: Wire Format Protocol. This is the document for the EMSMDB RPC interface.
- MS-OXCROPS: Remote Operations List and Encoding Protocol.

31.6 See also

`gromox(7)`, `http(8gx)`

EXCHANGE_NSP

32.1 Name

`exchange_nsp(4gx)` — `http(8gx)` processing plugin for the Exchange Server Name Service Provider Interface Protocol.

32.2 Description

`exchange_nsp` is a processing plugin for `http(8gx)` which handles the Exchange Server Name Service Provider Interface Protocol, in essence providing the Address Book for the EMSMDB connector.

32.3 Config file directives

`cache_interval` Default: *5 minutes*

`hash_table_size` Default: *3000*

`max_item_num` Default: *100000*

`session_check` Default: *false*

`x500_org_name` Default: (unspecified)

32.4 Notes

A number of properties are always synthesized by `exchange_nsp` and never read from any storage; this includes key properties such as `PR_ENTRYID`, `PR_RECORD_KEY`, etc. The following properties need mentioning:

- `PR_DEPARTMENT`: This property is synthesized from the department that a user has been assigned to (cf. SQL table “users”, column “group_id”, and SQL table “group”, column “title”).

For user-attached properties that are read from SQL (cf. table “user_properties”), `exchange_nsp` (as well as `zcore(8gx)`’s AB) only handles a subset of property types: `PT_BOOLEAN`, `PT_SHORT`, `PT_LONG`, `PT_I8`, `PT_SYSTIME`, `PT_BINARY`, `PT_UNICODE/STRING8` and `PT_MV_UNICODE/STRING8`.

32.5 Normative references

- MS-OXNSPI: Exchange Server Name Service Provider Interface (NSPI) Protocol

32.6 See also

gromox(7), **http(8gx)**

EXCHANGE_RFR

33.1 Name

`exchange_rfr(4gx)` — `http(8gx)` processing plugin for the Address Book Name Service Provider Interface Referral Protocol.

33.2 Description

`exchange_rfr` is a processing plugin for `http(8gx)` which handles the Address Book Name Service Provider Interface Referral Protocol.

33.3 Configuration file directives

This plugin has no directives.

33.4 Normative references

- MS-OXABREF: Address Book Name Service Provider Interface (NSPI) Referral Protocol

33.5 See also

`gromox(7)`, `http(8gx)`

EXMDB_LOCAL

34.1 Name

`exmdb_local(4gx)` — LDA hook plugin that offers a `exmdb_provider(4gx)` client with a C API

34.2 Description

An LDA hook plugin for `delivery(8gx)` which places mail into a store by connecting to a `exmdb_provider(4gx)` service.

34.3 Config file directives

This plugin has no directives.

34.4 Files

- `data_file_path/propnames.txt`
- `config_file_path/exmdb_list.txt`: `exmdb` multiserver selection map, see `exmdb_provider(4gx)` for details.
- `data_file_path/local_bounce/`: response templates for when mail cannot be delivered

34.5 See also

`gromox(7)`, `delivery(8gx)`, `exmdb_provider(4gx)`

EXMDB_PROVIDER

35.1 Name

exmdb_provider(4gx) — http(8gx) service plugin for exmdb connections

35.2 Description

exmdb_provider is a service plugin for http(8gx). It offers a plethora of individual functions (124 of them) for operating on mailbox stores. In addition, this functionality is also exposed by way of a Gromox-specific network protocol on port 5000.

35.3 Configuration file directives

cache_interval Default: *2 hours*

enable_dam

When set to **on**, inbox rule processing is allowed to create Deferred Action Messages (DAM). Furthermore, the “Deferred Actions” folder will have its contents shown. / Conversely, if this directive is **off**, no DAMs will be created, and the DAM folder in inboxes is presented as empty to clients (even if it has content from earlier).

Outlook’s DAM handling is poor and if you experience a crash with a primary mailbox that is in non-cached/online mode a few seconds after Outlook has opened it, turn this option off for mitigation.

Default: *on*

exrpc_debug

Log every incoming exmdb network RPC and the return code of the operation in a minimal fashion to stderr. Level 1 emits RPCs with a failure return code, level 2 emits all RPCs. Note that direct function calls from within the process image are not logged this way, so this will not show exmdb_provider invocations from exchange_emsmdb(4gx).

Default: *0*

listen_ip

An IPv6 address (or v4-mapped address) for exposing the timer service on.

Default: *::1*

listen_port

The TCP port number for exposing the timer service on.

Default: *5000*

max_ext_rule_number Default: *20*

max_router_connections Default: unlimited

max_rpc_stub_threads Default: unlimited

max_rule_number Default: 1000

max_store_message_count Default: 200000

notify_stub_threads_num Default: 4

populating_threads_num Default: 4

rpc_proxy_connection_num Default: 10

separator_for_bounce Default: ;

sqlite_mmap_size

See https://www.sqlite.org/pragmas.html#pragma_mmap_size for details.

Default: 0 (use SQLite default)

sqlite_synchronous

Enables/disables synchronous mode for SQLite databases. See https://www.sqlite.org/pragmas.html#pragma_synchronous for details.

Default: *off*

sqlite_wal_mode

Selects the particular journal mode for SQLite databases; **off** selects DELETE mode, **on** selects WAL mode. See https://www.sqlite.org/pragmas.html#pragma_journal_mode for details.

Default: *on*

table_size Default: 5000

x500_org_name

Default: (unspecified)

35.4 Multiserver selection map

The SQL column **users.homedir** specifies a home directory location in an abstract namespace. This abstract namespace is shared between all Gromox programs, and can be used to divide users into custom subsets and steer connections to different servers.

`exmdb_list.txt` specifies how to map from this namespace to `exmdb` servers. The file is used by `exmdb` clients to select the right server to connect to, and the file is used by `exmdb_provider` to set up its own data structures.

Each line in this file consists of 4 columns separated by whitespace:

- Initial prefix to match a user's `exmdb` home directory on. The pattern should almost always end in a `'/'` character, otherwise a prefix of `"/home"` is able to match a `userdir` of `"/home2/username"` as well, which may be undesired.
- The type of mail stores that are served beneath the prefix. This must either be `"private"` or `"public"`.
- The IPv6 (or v4-mapped) address of the `midb` server to use for this prefix.
- The port number.

In the absence of `exmdb_list.txt`, two implicit default entries are used:

```
/var/lib/gromox/user/ private ::1 5000
/var/lib/gromox/domain/ public ::1 5000
```

35.5 Network protocol

The transmissions on the socket are simple concatenations of protocol data units built using the NDR format. The PDU length is present within the PDU itself near the start.

```
{
    leuint32_t length;
    char pdu[];
}
```

```
pdu := {
    uint8_t call_id;
    string directory;
    switch (call_id) {
        ...
    }
}
```

35.6 Files

- *config_file_path/exmdb_acl.txt*: A file with one address (IPv6 or v4-mapped) per line of allowed clients. In its absence, `::1` is default-whitelisted.
- *config_file_path/exmdb_list.txt*: exmdb multiserver selection map.
- *data_file_path/mail_bounce/*

config_file_path and *data_file_path* is determined by the configuration of the program that loaded the *exmdb_provider* plugin.

35.7 See also

gromox(7), **http(8gx)**

36.1 Name

freebusy — Helper program for EWS Freebusy actions

36.2 Description

The freebusy program is used by the EWS handler (a PHP script) to implement the GetUserAvailabilityRequest request.

36.3 Files

- /usr/share/gromox/agent/exmdb_list.txt: exmdb multiserver map

36.4 See also

gromox(7)

GROMOX-ABKTCONV

37.1 Name

`gromox-abktconv` — Utility for converting between ABKT and JSON

37.2 Synopsis

```
gromox-abktconv {-b|-j} [-gw?] [-c cpid]
```

37.3 Description

`gromox-abktconv` can be used to convert between data streams as specified in MS-OXOABKT and a textual representation. It reads and writes to standard input and output, respectively.

37.4 Options

- b** Produce type-1 ABKT from JSON.
- c *cpid*** When converting to ABKT (-b), convert strings to the given codepage and emit them as 8-bit strings.
- g** When converting to ABKT (-b), emit extraneous gaps in the data stream to mimic what Exchange would do. Without -g, the ABKT stream will have no unnecessary gaps.
- j** Produce JSON from ABKT type-1 or type-2.
- w** When converting to ABKT (-b), emit strings in UTF-16 form. This is the default.
- ?** Display option summary.

37.5 Normative references

- MS-OXOABKT: Address Book User Interface Templates Protocol
- `oxoabkt.rst`: Type-2 ABKT template protocol

37.6 See also

`gromox(7)`, `gromox-abktpull(8gx)`

GROMOX-ABKTPULL

38.1 Name

`gromox-abktpull` — Utility to extract ABKT templates from LDIF

38.2 Synopsis

`ldapsearch ... | gromox-abktpull`

38.3 Description

`gromox-abktpull` reads LDIF from standard input and extracts the values from attributes matching `/DisplayTable/` — i.e. generally `originalDisplayTable`, `originalDisplayTableMSDOS`, `addressEntryDisplayTable` and `addressEntryDisplayTableMSDOS` — and saves them in separate files in the current working directory.

38.4 Normative references

- RFC 2849: LDAP Data Interchange Format

38.5 See also

`gromox(7)`, `gromox-abktconv(8gx)`

GROMOX-DBOP

39.1 Name

`gromox-dbop` — Database maintenance utility

39.2 Synopsis

`gromox-dbop` [-CU] [-create-old] [-c *mysql_adaptor.cfg*]

39.3 Options

- C Create the initial set of tables for the user information database (in MySQL).
- U Upgrade the schema of the user information database. The database (MYSQL_DBNAME) should exist and be blank.
- create-old Create blank initial database using version n0. (This can be used for testing.)
- c *mysql_adaptor.cfg* Path to a configuration file that defines MYSQL_HOST, MYSQL_USERNAME, etc. If omitted, `gromox-dbop` will read `/etc/gromox/http.cfg` to locate `mysql_adaptor.cfg` to locate the `mysql` parameter.

39.4 See also

`gromox(7)`, `mysql_adaptor(4gx)`

GROMOX-KDB2MT

40.1 Name

gromox-kdb2mt — Utility for analysis of Kopano databases

40.2 Synopsis

gromox-kdb2mt [-pstv] [-src-host *sqlhost*] [-src-user *sqluser*] [-src-port *sqlport*] --src-at *dir* {-src-guid *mbxguid*|-src-mbox *username*} [...]

40.3 Description

gromox-kdb2mt reads one store from a Zarafa/Kopano SQL database and, for attachments, the associated filesystem. The data is then re-exported in a Gromox-specific mailbox transfer format to stdout for consumption by pipe by the **gromox-mt2exm(8gx)** program. Optionally, **kdb2mt** can print a summary of the hierarchy during extraction.

The SQL server that carries the Zarafa/Kopano database for the home server of the user must be active. Databases with a schema version **n61** or newer (ZCP 7.0.3 and onwards, all KC) are supported.

The **kopano-server(8)** process need not be running.

The directory of Kopano attachments of the home server should be made available in the mount namespace wherever **gromox-kdb2mt** runs.

40.4 Options

- p Show properties in detail (enhances -t).
- s Splice objects from the source database into existing (Gromox-side) folders. Specifically, the source mailbox root is mapped to the Gromox store root, and special folders (Sent Items, Deleted Items, etc.) are mapped to special folders in the target store.
- t Show a diagnostic tree view of the source data as it is being read.
- v Print message count progress while processing larger folders. This option has no effect if (the even more verbose) -t option was used.

--src-host *hostname*

Hostname for the source SQL connection.

Default: (MySQL default; *localhost*)

--src-port *port*

Port for the source SQL connection.
Default: (MySQL default; automatic)

-src-user *username*

Username for the source SQL connection.
Default: *root*

-src-at *directory* Required specification to the `/var/lib/kopano/attachments` directory mounted somewhere locally. (To skip over file-based attachments, use the empty value, i.e. **-src-at ""**.)

-src-guid *guid* Extract the mailbox with the particular GUID from the source. (May be used to read orphaned stores.)

-src-mbox *username* Extract the mailbox of the given username from the source.

-l1 *x*, -l2 *y* If you are using `attachment_storage=files_v1-x-y` in `kopano-server.cfg`, call `kdb2mt` with the L1 and L2 options.

-only-obj *hid* Extract just the object with the given hierarchy id. This option may be specified multiple times. In the output stream, the objects will have an unspecified location given as their parent folder. (`mt2exm` will use the drafts folder because it is expectedly less crowded than one's inbox.)

40.5 Environment variables

SRCPASS Password for the source SQL connection.

40.6 Examples

Scenario 1. Gromox and Kopano are located on two different hosts; `gromox-kdb2mt` is executed from the Gromox system and pulls from a remote SQL and remote filesystem presented locally.

```
sshfs root@kdb.lan:/var/lib/kopano/attachments /tmp/at
```

```
SRCPASS=rootpass gromox-kdb2mt -src-sql kdb.lan -src-attach /tmp/at -src-mbox jdoe -dst-mbox john.doe@gromox.lan | gromox-mt2exm target@domain.de
```

Scenario 2. Gromox can be installed on the Kopano host, pulling from a local/remote SQL and local filesystem, while sending its export stream to another machine.

```
gromox-kdb2mt -src-attach /var/lib/kopano/attachments -src-mbox jdoe | ssh -Tenone root@gromox.lan "gromox-mt2exm -u target@domain.de"
```

It is also possible to copy the attachment directory beforehand, or to store the MT data stream, though additional disk space for these intermediate objects and representation would be required.

40.7 See also

`gromox(7)`, `gromox-mt2exm(8gx)`

GROMOX-MAILQ

41.1 Name

`gromox-mailq` — Local delivery agent queue lister

41.2 Synopsis

`gromox-mailq`

41.3 Description

`gromox-mailq` looks in the queue directory for `delivery-queue(8gx)` and prints the summaries for all queued mail.

41.4 See also

`gromox(7)`, `delivery-queue(8gx)`

GROMOX-MKMIDB

42.1 Name

`gromox-mkmidb` — Tool for creating a blank message index database

42.2 Synopsis

`gromox-mkmidb` [-c *config*] [-d *datapath*] *username*

42.3 Options

-c *config* Read configuration directives from the given file. If this option is not specified, `/etc/gromox/sa.cfg` will be read if it exists.

-d *datapath* This option can be used to override the `data_file_path` variable from the config file.

-? Display option summary.

42.4 Files

- `data_file_path/sqlite3_midb.txt`: SQLite instructions to generate a message index database.

42.5 See also

`gromox(7)`, `sa.cfg(5gx)`

GROMOX-MKPRIVATE

43.1 Name

`gromox-mkprivate` — Tool for creating a blank private store

43.2 Synopsis

`gromox-mkprivate` [-c *config*] [-d *datapath*] *username*

43.3 Options

-c *config* Read configuration directives from the given file. If this option is not specified, `/etc/gromox/sa.cfg` will be read if it exists.

-d *datapath* This option can be used to override the `data_file_path` variable from the config file.

-? Display option summary.

43.4 Files

- `data_file_path/folder_lang.txt`: Translations for essential folders in a message store.
- `data_file_path/proppnames.txt`: Initial set of named properties to add to the new private store.
- `data_file_path/sqlite3_common.txt`: SQLite instructions to generate part of a private store.
- `data_file_path/sqlite3_private.txt`: SQLite instructions to generate part of a private store.

43.5 Config file directives

The config file is shared with other programs. See `sa.cfg(5gx)`.

43.6 See also

gromox(7), **gromox-mkmidb(8gx)**, **gromox-mkpublic(8gx)**, **sa.cfg(5gx)**

GROMOX-MKPUBLIC

44.1 Name

`gromox-mkpublic` — Tool for creating a blank public store

44.2 Synopsis

gromox-mkpublic [-c *config*] [-d *data_path*] *domainname*

44.3 Options

-c *config* Read configuration directives from the given file. If this option is not specified, `/etc/gromox/sa.cfg` will be read if it exists.

-d *datapath* This option can be used to override the `data_file_path` variable from the config file.

-? Display option summary.

44.4 Files

- `data_file_path/propnames.txt`: Initial set of named properties to add to the new private store.
- `data_file_path/sqlite3_common.txt`: SQLite instructions to generate part of a public store.
- `data_file_path/sqlite3_public.txt`: SQLite instructions to generate part of a public store.

44.5 Config file directives

The config file is shared with other programs. See `sa.cfg(5gx)`.

44.6 See also

`gromox(7)`, `gromox-mkprivate(8gx)`, `sa.cfg(5gx)`

GROMOX-MT2EXM

45.1 Name

`gromox-mt2exm(8gx)` — Utility for importing various mail items

45.2 Synopsis

```
gromox-mt2exm [-pt] -u [user]@domain.de
```

45.3 Description

`gromox-mt2exm` reads a Gromox-specific mailbox transfer format data stream from standard input as generated by `gromox-pff2mt(8gx)` and writes folders and messages included therein to a specific Gromox mail store.

When importing to a public folder, the message owner will be the system account and no user can delete/modify these messages.

45.4 Options

- `-p` Show properties in detail (enhances `-t`).
- `-t` Print folder/message summary as these items are processed.
- `-u [user]@domain.de` Target mail store which to import mails to. For the public folder of a domain, leave out the local part, i.e. use `@domain.de`.

45.5 Examples

PFF import to a private mailbox:

```
gromox-pff2mt sample.pst | gromox-mt2exm -u target@mailbox.de
```

PFF import to a public folder:

```
gromox-pff2mt sample.pst | gromox-mt2exm -u @mailbox.de
```

45.6 See also

`gromox(7)`, `gromox-pff2mt(8gx)`

GROMOX-PFF2MT

46.1 Name

`gromox-pff2mt(8gx)` — Utility for analysis of PFF/PST/OST files

46.2 Synopsis

`gromox-pff2mt [-pst] input.pst`

46.3 Description

`gromox-pff2mt` reads a file that conforms to the Personal Folder File (PFF) and the Offline Folder File (OFF) format and re-exports the data in a Gromox-specific mailbox transfer format to stdout for consumption by pipe by the `gromox-mt2exm(8gx)` program. Optionally, `pff2mt` can print a tree summary of the PFF. The PFF format is used in several file types:

- PAB (Personal Address Book)
- PST (Personal Storage Table)
- OST (Offline Storage Table)

By default, `pff2mt` creates a new folder “Import of <xyz.pst> on <date>” within `IPM_SUBTREE` (“Top of Information Store”) and places all PFF objects into that new subfolder.

46.4 Options

- p Show properties in detail (enhances -t).
- s Splice objects from the PFF into existing folders. Specifically, the PFF root is mapped to the store root, and special folders (Sent Items, Deleted Items, etc.) are mapped to special folders in the store if the PFF has the necessary information.
- t Show a diagnostic tree view of the source data as it is being read.

46.5 Examples

Import of a PFF-compatible file into a Gromox mailbox via exmdb transport:

```
gromox-pff2mt sample.pst | gromox-mt2exm -u target@mailbox.de
```

46.6 See also

[gromox\(7\)](#), [gromox-mt2exm\(8gx\)](#)

47.1 Name

`gromox-pffimport(8gx)` — Utility for importing PFF/PST/OST into Gromox

47.2 Synopsis

```
gromox-pff2mt [-s] [-pt] input.pst | gromox-mt2exm -u mbox@target.de
```

47.3 Description

The `gromox-pffimport` utility has been replaced by separate programs intended to be connected via pipe. This allows for a more varied set of conversion formats.

47.4 See also

`gromox(7)`, `gromox-mt2exm(8gx)`, `gromox-pff2mt(8gx)`

GROMOX-REBUILD

48.1 Name

`gromox-rebuild` — Tool for rebuilding a message store

48.2 Synopsis

`gromox-rebuild` [-c *config*] [-d *datapath*] *maildir*

48.3 Options

-c *config* Read configuration directives from the given file. If this option is not specified, `/etc/gromox/sa.cfg` will be read if it exists.

-d *datapath* This option can be used to override the `data_file_path` variable from the config file.

48.4 Files

- `config_file_path/exmdb_list.txt`: exmdb multiserver selection map.
- `data_file_path/sqlite3_common.txt`: SQLite instructions to generate part of a private store.
- `data_file_path/sqlite3_private.txt`: SQLite instructions to generate part of a private store.

48.5 Config file directives

The config file is shared with other programs. See `sa.cfg(5gx)`.

48.6 See also

`gromox(7)`, `sa.cfg(5gx)`

49.1 Name

http — Central server for RPC and the message database

49.2 Synopsis

http [-c *config*]

49.3 Description

http(8gx) is a trivial HTTP server. It understands the special HTTP methods as used by RPC-over-HTTP protocol as used by Outlook, it can serve files verbatim, or forward requests to a FastCGI server such as php-fpm(8).

49.4 Options

-c *config* Read configuration directives from the given file. If this option is not specified, /etc/gromox/http.cfg will be read if it exists.

-? Display option summary.

49.5 URI processing order

- Requests are passed to the mod_rewrite(4gx) module (built-in) to have their URI potentially rewritten.
- If a HTTP request is using the methods RPC_IN_DATA or RPC_OUT_DATA, the data stream is handed off to the exchange_emsmb(4gx) plugin.
- Otherwise, HPM plugins are invoked. Processing ends when one plugin signals that the request was handled. The order depends on the HPM plugin list (cf. http.cfg(5gx):**hpm_plugin_list**).
- Otherwise, the mod_fastcgi(4gx) module (built-in) is invoked. Processing ends if the module handled the request.
- Otherwise, the mod_cache(4gx) module (built-in) is invoked. Processing ends if the module handled the request.
- Otherwise, the request is rejected.

49.6 RPC-over-HTTP

RPC-over-HTTP utilizes two special HTTP methods, `RPC_IN_DATA` and `RPC_OUT_DATA`. These requests can, similarly to HTTP `CONNECT`, be very long-lived. The RPC data stream is handled by the included `exchange_emsmb(4gx)` plugin.

49.7 Configuration file directives

block_interval_auths

The amount of time a user is blocked from connecting to the service after too many failed logins.
Default: *1 minute*

config_file_path

Colon-separated list of directories which will be scanned when locating further configuration files, especially those used by plugin instances.
Default: */etc/gromox/http:/etc/gromox*

console_server_ip

An IPv6 address (or v4-mapped address) to expose the management console frontend on.
Default: *::1*

console_server_port

The TCP port number to expose the management console frontend on.
Default: *8899*

context_average_mem Default: *256K*

context_num Default: *400*

data_file_path

Colon-separated list of directories which will be scanned when locating data files.
Default: */usr/share/gromox/http*

default_domain Default: (inherited from system)

fastcgi_cache_size

If the HTTP request to a CGI endpoint uses Chunked Transfer Encoding and the Content-Length field contains a larger value than this limit, the data is buffered in a file */tmp/http-%d* (%d replaced by internal context id).
Default: *256K*

fastcgi_exec_timeout

Maximum execution time for CGI scripts.
Default: *10 minutes*

fastcgi_max_size

If the Content-Length of a HTTP request to a CGI endpoint is larger than this value, the request is rejected.
Default: *4M*

host_id

The hostname that the server uses to identify itself (e.g. Server: header in HTTP responses).
Default: (inherited from system)

hpm_cache_size

If the HTTP request to a HPM endpoint uses Chunked Transfer Encoding and the Content-Length field contains a larger value than this limit, the data is buffered in a file `/tmp/http-%d` (%d replaced by internal context id).

Default: *512K*

hpm_max_size

If the Content-Length of a HTTP request to a HPM endpoint is larger than this value, the request is rejected.

Default: *4M*

hpm_plugin_ignore_errors

If set to yes, HPM plugins that fail to load on startup are ignored. If set to no, the daemon will exit if any plugin cannot be loaded.

Default: *no*

hpm_plugin_list

Path to a text file which lists the filenames of HPM plugins to load, one per line.

Default: (unspecified)

http_auth_times

The number of login tries a user is allowed before the account is blocked.

Default: *10*

http_certificate_passwd

The password to unlock TLS certificates.

Default: (unset)

http_certificate_path

Filesystem path to a certificate file for use with encrypted connection. The complete certificate chain should be present (as there is no other config directive to pull CA certs in).

Default: (unset)

http_conn_timeout

If a HTTP connection stalls for the given period, the connection is terminated.

Default: *3 minutes*

http_debug

If set to **1**, prints all incoming and outgoing HTTP traffic to stderr.

Default: *0*

http_private_key_path

Filesystem path to the key file needed to unlock the TLS certificate.

Default: (unset)

http_support_ssl

This flag controls whether (or not) the server offers TLS at all. The default is false because you need a certificate for this first.

Default: *false*

listen_port

The TCP port to expose the HTTP protocol service on. (The IP address is fixed to the wildcard address.)

Default: *80*

listen_ssl_port

The TCP port to expose implicit-TLS HTTP protocol service (HTTPS) on. (The IP address is fixed to the wildcard address.)

Default: (unset)

proc_plugin_ignore_errors

If set to yes, processor plugins that fail to load on startup are ignored. If set to no, the daemon will exit if any plugin cannot be loaded.

Default: *no*

proc_plugin_list

Path to a text file which lists the filenames of processor plugins to load, one per line.

Default: (unspecified)

proc_plugin_path

Filesystem path for processor plugins.

Default: */usr/libexec/gromox*

request_max_mem The maximum hint size for fragmented RPC PDU requests that will be allowed (C706 §12.6.3.7, RPCE §2.2.2.6).

running_identity

An unprivileged user account to switch the process to after startup. To inhibit the switch, assign the empty value.

Default: *gromox*

service_plugin_ignore_errors

If set to yes, service plugins that fail to load on startup are ignored. If set to no, the daemon will exit if any plugin cannot be loaded.

Default: *no*

service_plugin_list

Path to a text file which lists the filenames of service plugins to load, one per line.

Default: (unspecified)

service_plugin_path

Path to a secondary directory where service plugins will be loaded from if a primary search in standard directories (as per `ld.so(8)`) was unsuccessful.

Default: */usr/lib/gromox*

state_path

Directory for runtime variadic data.

Default: */var/lib/gromox*

tcp_mss_size

Sets the TCP_MAXSEG socket option with the given MSS value for the listening socket(s), cf. `tcp(7)`.

Default: *0* (do not limit the MSS)

thread_charge_num

The maximum number of connections that each thread is allowed to process.

Default: *20*

thread_init_num

The minimum number of client processing threads to keep around.

Default: *5*

user_default_lang Default: *en*

49.8 Files

- /usr/lib/gromox/libgxm_*.so: HTTP processing plugins
- /usr/lib/gromox/libgxp_*.so: PDU processing plugins
- /usr/lib/gromox/libgxs_*.so: service plugins

49.9 Normative references

- MS-RPCE: Remote Procedure Call Protocol Extensions
- DCERPC / C706: Technical Standard DCE 1.1: Remote Procedure Call by The Open Group, 1997

49.10 See also

gromox(7), **mod_cache(4gx)**, **mod_fastcgi(4gx)**, **mod_rewrite(4gx)**

50.1 Name

imap — Gromox IMAP server

50.2 Synopsis

imap [-c *config*]

50.3 Options

-c *config* Read configuration directives from the given file. If this option is not specified, `/etc/gromox/imap.cfg` will be read if it exists.

-version Output version information and exit.

-? Display option summary.

50.4 Configuration file directives

block_interval_auths

The amount of time a user is blocked from connecting to the service after too many failed logins.

Default: *1 minute*

config_file_path

Colon-separated list of directories in which further configuration files, especially those used by plugin instances, will be searched.

Default: */etc/gromox/imap:/etc/gromox*

console_server_ip

An IPv6 address (or v4-mapped address) to expose the management console frontend on.

Default: *::1*

console_server_port

The TCP port number to expose the management console frontend on.

Default: *4455*

context_average_mem Default: *128K*

context_average_mitem

Lower clamp is at 128.

Default: *512*

context_max_mem Default: *2M*

context_num

Maximum number of concurrently active sessions.

Default: *200*

data_file_path

Colon-separated list of directories in which static data files will be searched.

Default: */usr/share/gromox/imap*

default_domain Default: (system domainname)

default_lang Default: *en*

enable_rfc2971_commands

RFC 2971 specifies the “ID” command with which a client can inquire the program name and version of the server. This is disabled by default, as it can facilitate potential attackers’ information gathering.

Default: *no*

host_id Default: (system hostname)

imap_auth_times

The number of login tries a user is allowed before the account is blocked.

Default: *10*

imap_autologout_time

If an IMAP connection is idle for the given period, the connection is terminated.

Default: *30 minutes*

imap_certificate_passwd

The password to unlock TLS certificates.

Default: (unset)

imap_certificate_path

Filesystem path to a certificate file for use with encrypted connection. The complete certificate chain should be present (as there is no other config directive to pull CA certs in).

Default: (unset)

imap_conn_timeout

If an IMAP connection stalls for the given period, the connection is terminated.

Default: *3 minutes*

imap_force_starttls

This flag controls whether clients must utilize TLS, either by way of implicit TLS (cf. **listen_ssl_port**), or through the STARTTLS command.

Default: *false*

imap_lang_path

Path to the file for IMAP folder name translations. If only a basename is specified, the file will be searched in *data_file_path*.

Default: *imap_lang.txt*

imap_private_key_path

Filesystem path to the key file needed to unlock the TLS certificate.

Default: (unset)

imap_support_starttls

This flag controls the offering of the STARTTLS extension (RFC 2595) to clients.

Default: *false*

listen_port

The TCP port to expose the IMAP protocol service on. (The IP address is fixed to the wildcard address.)

Default: *143*

listen_ssl_port

The TCP port to expose implicit-TLS IMAP protocol service (IMAPS) on. (The IP address is fixed to the wildcard address.)

Default: (unset)

running_identity

An unprivileged user account to switch the process to after startup.

Default: *gromox*

service_plugin_ignore_errors

If set to yes, service plugins that fail to load on startup are ignored. If set to no, the daemon will exit if any plugin cannot be loaded.

Default: *no*

service_plugin_list

Path to a text file which lists the filenames of service plugins to load, one per line.

Default: (unspecified)

service_plugin_path

Path to a secondary directory where service plugins will be loaded from if a primary search in standard directories (as per ld.so(8)) was unsuccessful.

Default: */usr/lib/gromox*

state_path

Directory for runtime variadic data.

Default: */var/lib/gromox*

thread_charge_num

The maximum number of connections that each thread is allowed to process.

Default: *40*

thread_init_num

The minimum number of client processing threads to keep around.

Default: *1*

50.5 Files

- *data_file_path/imap_code.txt*: Mapping from internal IMAP error codes to textual descriptions.
- *data_file_path/imap_lang.txt*: Translations for IMAP folder names.
- */usr/lib/gromox/libgxs_*.so*: service plugins

50.6 See also

gromox(7), **midb_agent(4gx)**

IP6_CONTAINER

51.1 Name

ip6_container — trivial source connection counter

51.2 Description

ip6_container is used by one or more daemons to keep track of the number of connections made by every source address, and to block them from exceeding a connection limit.

51.3 Configuration file directives

connection_max_num

The maximum number of concurrent connections any one source address is allowed to make.

Default: *200*

51.4 See also

gromox(7)

LDAP_ADAPTOR

52.1 Name

ldap_adaptor — LDAP connector for authentication

52.2 Description

ldap_adaptor is a service plugin for Gromox that facilitates the use of an LDAP server for authentication purposes. Since the authoritative user database is in MySQL, LDAP is only used to perform authentication (Bind operations), and metadata searches that lead up to such Binds, i.e. looking for the LDAP object that has a particular “mail” attribute.

52.3 Configuration file directives

auth_connections

The maximum number of LDAP connections that may be used for authentication requests. (Authentication cannot be run on the metadata connections, and so this constitutes a separate set of connections.) This parameter has fixed value and is currently not settable.

Default: (same as data_connections)

data_connections

The number of LDAP connections that will be kept active to the LDAP server for the purpose of metadata searches.

Default: 4

ldap_bind_user

An LDAP binddn to use for metadata searches. You cannot use SASL mechanisms/SASL users at this time.

Default: (unset)

ldap_bind_pass

Password for Simple Authentication of ldap_bind_user.

Default: (unset)

ldap_host

Whitespace-separated RFC 2255-style set of LDAP URIs that specify protocol and host.

Default: (libldap default, see ldap.conf(5))

ldap_mail_attr Default: *mail*

ldap_search_base Default: (libldap default)

ldap_start_tls

Use the STARTTLS mechanism on LDAP connections.

Default: *off*

52.4 See also

gromox(7), **authmgr(4gx)**

53.1 Name

logthru — service plugin for a stdout/file logger

53.2 Description

logthru registers a “log_info” service function. Log messages sent to this API will be emitted without delay to stdout or a file of choice. When programs are run under the control of systemd (i.e. have been started with systemctl), journald will take care of capture, storage and periodic log rotation.

logthru is the default logging plugin for all Gromox services.

53.3 Configuration file directives

log_file_name

Path to the file where log messages will be appended to. Setting this to the empty value implies that logging will go to stdout.

Default: (stdout)

log_level

Controls which messages will be shown. Messages with a equal or higher severity (lower numeric value) will pass, messages with lower severity (higher numeric value) will be suppressed. The levels used in practice are: critical (2), error (3), warning (4), notice/info (5), debug (6).

Default: 4

53.4 See also

gromox(7)

54.1 Name

mapi.so — PHP module providing MAPI functions

54.2 Description

mapi.so is a PHP module that makes available a number of functions to PHP for connecting to Gromox services. In particular, it will talk to zcore(8gx).

54.3 Configuration

The PHP ini fragment, mapi.ini, may look like this:

```
extension=mapi.so
[mapi]
zcore_socket=/run/gromox/zcore.sock
```

54.4 Compatibility with other implementations

The Gromox implementation of mapi.so is missing (-) some functions and offering others (+) compared to KGWC:

```
-mapi_createconversationindex
-mapi_folder_openmodifytable
-mapi_freebusydata_enumblocks
-mapi_freebusydata_getpublishrange
-mapi_freebusydata_setrange
-mapi_freebusyenumblock_ical
-mapi_freebusyenumblock_next
-mapi_freebusyenumblock_reset
-mapi_freebusyenumblock_restrict
-mapi_freebusyenumblock_skip
-mapi_freebusy_openmsg
-mapi_freebusysupport_close
-mapi_freebusysupport_loaddata
-mapi_freebusysupport_loadupdate
-mapi_freebusysupport_open
-mapi_freebusyupdate_publish
```

(continues on next page)

(continued from previous page)

```
-mapi_freebusyupdate_reset  
-mapi_freebusyupdate_savechanges  
-mapi_icaltomapi2  
-mapi_importcontentschanges_importmessagechange  
-mapi_rules_gettable  
-mapi_rules_modifytable  
-mapi_vcftomapi2  
-mapi_zarafa_getcompanylist  
-mapi_zarafa_getgrouplist  
-mapi_zarafa_getgrouplistofuser  
-mapi_zarafa_getquota  
-mapi_zarafa_getuser_by_id  
-mapi_zarafa_getuser_by_name  
-mapi_zarafa_getuserlist  
-mapi_zarafa_getuserlistofgroup  
-mapi_zarafa_setquota  
+mapi_folder_getrulestable  
+mapi_folder_modifyrules  
+mapi_getuseravailability  
+mapi_importcontentschanges_importmessagechange  
+mapi_linkmessage  
+mapi_logon_ex  
+mapi_openpropertytostream  
+mapi_table_createbookmark  
+mapi_table_findrow  
+mapi_table_freebookmark  
+nsp_getuserinfo  
+nsp_setuserpasswd
```

54.5 See also

gromox(7), **zcore(8gx)**

55.1 Name

map — Definition for “Messaging Application Programming Interface”

55.2 Definition

MAPI (Messaging API) can have many different meanings. As abstracta:

- The concept of the hierarchial organization into **containers**, **tables**, **contents** and **properties**. Typical containers are **stores** and **folders**, typical contents are **messages**, **recipients**, **attachments**.
- More concrete variations, such as the set of “well-known” properties like, for example, subject, modification date, body, etc., and the semantics expected of them.
- Optionally, the profile configuration structure present in some MAPI client implementations, comprised of **profiles**, **providers**, **services**, **profile sections**, **sessions** and their associated properties. A server generally does not know or need to know about these.
- A specification of types and functions to make use of any of these concepts in a programming language.

As an implementation or part thereof:

- (Seemingly the most common meaning of “MAPI”!) The MSMAPI interface for programming in C and/or C++ that comes with the Windows operating system/Office package, with types like “IMAPIFolder” and “IMessage”, functions like “MAPILogonEx” or “IMessage::GetProperties”, and identifiers like PR_SUBJECT.
- The Office VBA interface for programming in VisualBasic, with types like “Outlook.MailItem”, object properties like “MailItem.Subject”.
- The wire protocols used for communication with Exchange Server. This has received unfortunate terms like “MAPI/RPC” or “MAPI protocol”. Specifically, this includes the independently specified parts of: the Remote Operations (ROP/OXCROPS) protocol, the EMSMDB v1 protocol, the NSPI protocol, the MAPI Extensions For HTTP protocol (MAPI-over-HTTP, MAPIHTTP), and their particular format of serialized representations of MAPI objects that each peer may have. Note that MAPIHTTP is not in any way “more MAPI” than RPC-over-HTTP (aka RPCH) already was: both MAPIHTTP and RPCH wrap OXCROPS.

55.3 Gromox implementation

Gromox does not offer an MSMAPI-style programming API for the C or C++ language environments; the types and functions that are exposed in .h/.hpp files are intended for its own use only and are not considered public let alone constant over time.

Gromox offers interfacing at the network level, e.g. via `exchange_emsmb(4gx)/http(8gx)` for the Exchange protocol formats, or via `exmdb_provider(4gx)` and `zcore(8gx)` using its own wire formats.

Gromox offers a somewhat MSMAPI-resembling interface for programming in PHP. This is a set of 119 freestanding functions that implement the concepts of stores, folders, messages, etc.

MESSAGE_ENQUEUE

56.1 Name

message_enqueue — Message enqueueing flusher plugin for delivery-queue(8gx)

56.2 Description

When the SMTP server has finished processing a mail, the mail will be handed off to a flusher plugin. message_enqueue is such a plugin, in fact, presently the only flusher plugin available.

message_enqueue(4gx) is the approximate equivalent of the Postfix cleanup(8) process.

56.3 See also

gromox(7), **delivery-queue(8gx)**

57.1 Name

mh_emsmb — MAPI-HTTP processor for EMSMDB

57.2 Description

mh_emsmb is a processing plugin for http(8gx) which reacts to URIs starting in `/mapi/emsmb/`. The ROP stream inside the HTTP request bodies is handed off to exchange_emsmb(4gx) for processing.

57.3 Config file directives

This plugin has no directives.

57.4 Normative references

- MS-OXCMAPIHTTP: Messaging Application Programming Interface (MAPI) Extensions for HTTP

57.5 See also

gromox(7), http(8gx), exchange_emsmb(4gx)

58.1 Name

mh_nsp — MAPI-HTTP processor for NSPI

58.2 Description

mh_nsp is a processing plugin for http(8gx) which reacts to URIs starting in **/mapi/nspi/**. The ROP stream inside the HTTP request bodies is handed off to exchange_nsp(4gx) for processing.

58.3 Config file directives

This plugin has no directives.

58.4 Normative references

- MS-OXCMAPIHTTP: Messaging Application Programming Interface (MAPI) Extensions for HTTP

58.5 See also

gromox(7), **http(8gx)**, **exchange_nsp(4gx)**

59.1 Name

midb — Message Index database daemon

59.2 Synopsis

midb [-c *config*]

59.3 Description

midb is used by imap(8gx) and pop3(8gx) to generate and keep track of folder and message summaries. In particular, it enumerates messages to give them their IMAP UID and have at all times a suitable UIDNEXT value for folders ready. midb also caches the Message-Id, modification date, message flags, subject and sender to facilitate IMAP listings.

59.4 Options

-c *config* Read configuration directives from the given file. If this option is not specified, `/etc/gromox/midb.cfg` will be read if it exists.

-version Output version information and exit.

-? Display option summary.

59.5 Configuration file directives

config_file_path

Colon-separated list of directories which will be scanned when locating further configuration files, especially those used by plugin instances.

Default: `/etc/gromox/midb:/etc/gromox`

console_server_ip

An IPv6 address (or v4-mapped address) to expose the management console frontend on.

Default: `::1`

console_server_port

The TCP port number to expose the management console frontend on.

Default: *9900*

data_file_path

Colon-separated list of directories which will be scanned when locating data files.

Default: */usr/share/gromox/midb*

default_charset Default: *windows-1252*

default_timezone Default: (unspecified)

midb_cache_interval Default: *30minutes*

midb_cmd_debug

Log every incoming MIDB command and the return code of the operation in a minimal fashion to stderr. Level 1 emits commands with a failure return code, level 2 emits all commands.

Default: *0*

midb_listen_ip

An IPv6 address (or v4-mapped address) for exposing the event service on.

Default: *::1*

midb_listen_port

The TCP port number for exposing the event service on.

Default: *5555*

midb_mime_number Default: *4096*

midb_table_size Default: *5000*

midb_threads_num

The minimum number of client processing threads to keep around.

Default: *100*

notify_stub_threads_num Default: *10*

rpc_proxy_connection_num

Default: *10*

service_plugin_list

Path to a text file which lists the filenames of service plugins to load, one per line.

Default: (unspecified)

service_plugin_path

Path to a secondary directory where service plugins will be loaded from if a primary search in standard directories (as per `ld.so(8)`) was unsuccessful.

Default: */usr/lib/gromox*

sqlite_mmap_size

See https://www.sqlite.org/pragmas.html#pragma_mmap_size for details.

Default: *0* (use SQLite default)

sqlite_synchronous

Enables/disables synchronous mode for SQLite databases. See https://www.sqlite.org/pragmas.html#pragma_synchronous for details.

Default: *off*

sqlite_wal_mode

Selects the particular journal mode for SQLite databases; **off** selects DELETE mode, **on** selects WAL mode. See https://www.sqlite.org/pragmas.html#pragma_journal_mode for details.

Default: *on*

state_path

Directory for runtime variadic data.

Default: */var/lib/gromox*

x500_org_name Default: (unspecified)

59.6 Files

- *config_file_path/exmdb_list.txt*: exmdb multiserver selection map.
- *config_file_path/midb_acl.txt*: A file with one address (IPv6 or v4-mapped) per line of allowed clients. In its absence, *:::1* is default-whitelisted.
- */usr/lib/gromox/libgxs_*.so*: service plugins

59.7 See also

gromox(7)

MIDB_AGENT

60.1 Name

midb_agent — Service plugin for conversing with midb(8gx)

60.2 Configuration file directives

connection_num

The number of connections to keep open towards every midb target.
Default: 5

context_average_mem

Enables and sets the size of a memory pool (in object count; the actual size is 256 bytes * context_num * context_average_mem). imap(8gx) and pop3(8gx) require this to be non-zero for full functionality.
Default: 1024

60.3 Multiserver map

The SQL column **users.homedir** specifies a home directory location in an abstract namespace. This abstract namespace is shared between all Gromox programs, and can be used to divide users into custom subsets and steer connections to different servers.

midb_list.txt specifies how to map from this namespace to midb servers. Each line in this file consists of 3 columns separated by whitespace:

- Initial prefix to match a user's exmdb home directory on. The pattern should almost always end in a '/' character, otherwise a prefix of "/home" is able to match a userdir of "/home2/username" as well, which may be undesired.
- The IPv6 (or v4-mapped) address of the midb server to use for this prefix.
- The port number.

In the absence of midb_list.txt, an implicit default entry is used:

```
/ ::1 5555
```


60.4 Files

- *config_file_path/instance.cfg*: configuration file for the instance of `midb_agent`. (Instance is usually **midb_agent**, as there is not much point in loading `midb_agent` twice.)
- *config_file_path/midb_list.txt*: `midb` multiserver map

config_file_path is determined by the configuration of the program that loaded the `midb_agent` plugin.

60.5 See also

`gromox(7)`

MOD_CACHE

61.1 Name

`mod_cache` — `http(8gx)` processing plugin for serving objects from a local filesystem

61.2 Description

`mod_cache` serves local files when certain URIs are requests.

`mod_cache` is built into `http(8gx)` and not a separate `.so` file.

61.3 Config file directives

This (built-in) plugin shares `http.cfg`. See `http(8gx)`.

61.4 URI map

The filemap that specifies which URIs to handle is `cache.txt`, which is searched for in `config_file_path`.

Each line in this file consists of 3 columns separated by whitespace:

- Domain or wildcard to match the HTTP Host: header with.
- URI path (prefix) to match
- Target directory within the filesystem

If the file has no lines, no documents will be served this way. If the file is absent however, a set of default entries will be used.

61.4.1 Default entries

```
* /web /usr/share/grommunio-web
```

61.5 Files

- *config_file_path/cache.txt*: URI map specifying which paths this plugin shall handle.

61.6 See also

gromox(7), **http(8gx)**

MOD_FASTCGI

62.1 Name

`mod_fastcgi` — `http(8gx)` processing plugin for proxying requests to FastCGI servers

62.2 Description

`mod_fastcgi` can forward HTTP requests to one or more FastCGI servers when certain URIs are requested. `mod_fastcgi` plugin is built into `http(8gx)` and not a separate `.so` file.

62.3 Config file directives

This (built-in) plugin shares `http.cfg`. See `http(8gx)`.

62.4 URI map

The filemap that specifies which URIs to handle is `fastcgi.txt`, which is searched for in `config_file_path`.

Each line in this file consists of 7 columns separated by whitespace:

- Domain or asterisk-based wildcard (“*”, “*.example.com”) to match the HTTP Host: request header with.
- URI path (prefix) to match
- Mapped path that will be passed to the FastCGI executor. Note that the FastCGI process may additionally have a document root setting that could map the path one more time.
- File suffix (without dot) to match on, e.g. *php*.
- A file to use as the default file for a directory (similar to Apache `httpd`’s `DirectoryIndex`).
- A set of headers, separated by the pipe symbol `|`, which should be forwarded to the CGI handler. In absence of any desired extra headers, a single pipe can be used to fill the column.
- An `AF_LOCAL` socket path to make the FastCGI request to.

If the file has no lines, no documents will be served this way. If the file is absent however, a set of default entries will be used.

62.4.1 Default entries

```
* /ews /usr/share/gromox/http/php/ews php index.php X-MAPIHttpCapability|X-
↳AnchorMailbox|X-ClientCanHandle /run/gromox/php-fpm.sock
* /sync /usr/share/grommunio-sync php index.php | /run/gromox/php-fpm.sock
* /web /usr/share/grommunio-web php index.php | /run/gromox/php-fpm.sock
```

62.5 Files

- *config_file_path/fastcgi.txt*: URI map specifying which paths this plugin shall handle.

62.6 See also

gromox(7), **http(8gx)**

MOD_REWRITE

63.1 Name

`mod_rewrite` — http(8gx) processing plugin for altering HTTP request URIs before processing

63.2 Description

`mod_rewrite` can alter request URIs. It runs before any of the other built-in or HPM processing plugins.

63.3 Config file directives

This (built-in) plugin shares `http.cfg`. See `http(8gx)`.

63.4 Rewrite map

The filemap that specifies which URIs to handle is `rewrite.txt`, which is searched for in `config_file_path`.

Each line in this file consists of 3 columns separated by whitespace:

- A POSIX Basic Regular Expression (cf. `regcomp(3)`) for matching the original URI.
- The fixed sequence “=>”.
- Replacement string. Captures can be spliced using `\1`, `\2`, .. up to a maximum of `\9`. The sequence `\0` splices the entire string (equivalent of Perl’s `$&`).

If the file has no lines, no paths will be rewritten. If the file is absent however, a set of default entries will be used.

63.5 Default rules

```
/autodiscover/autodiscover.xml => \0/ews/autodiscover.php
/EWS/Exchange.asmx => \0/ews/exchange.php
/OAB/oab.xml => \0/ews/oab.php
\(/Microsoft-Server-ActiveSync\) => \1/grommunio-sync/index.php
\(/.well-known/autoconfig/mail/config-v1.1.xml\) => \1/well-known/autoconfig-mail.php
```

63.6 Files

- *config_file_path/rewrite.txt*: Rewrite map specifying which paths this plugin shall handle.

63.7 See also

gromox(7), **http(8gx)**

MYSQL_ADAPTOR

64.1 Name

mysql_adaptor — MySQL/MariaDB connector for user metadata and authentication

64.2 Description

mysql_adaptor is a service plugin for integrating mail accounts from a MySQL/MariaDB database.

64.3 Configuration file directives

connection_num

Number of SQL connections to keep active.

Default: 8

enable_firsttimepw

This flag determines whether non-LDAP users with no recorded password (empty users.password SQL column) will have the account's password set to whatever credential was passed along in the first authentication request.

Default: o

mysql_dbname Default: *archive*

mysql_host

The hostname/IP address for contacting the SQL server.

Default: *localhost*

mysql_password Default: (unset)

mysql_port

The TCP port number for contacting the SQL server.

Default: 3306

mysql_rdwr_timeout

This sets the MYSQL_OPT_READ_TIMEOUT and MYSQL_OPT_WRITE_TIMEOUT option values on the MySQL connection.

Default: 0 (no timeout)

mysql_username Default: *root*

scan_interval Default: *1 minute*

schema_upgrades This controls what to do when a database schema update is available. Because there is no locking mechanism available that works across machines, the `mysql_adaptor` plugin does not use any locking locally either. As a result, the default is to not perform any upgrades.

autoupdate Do perform automatic schema upgrades when `mysql_adaptor` is started. Warning: This option can potentially be harmful if multiple programs try to upgrade the database at the same time.

host:xyz Perform automatic schema upgrades, but only when executing from `http(8gx)` and only when `http.cfg`'s **host_id** value matches `xyz`.

skip No automatic schema upgrades are performed. The initialization will report completion and the program continues execution. The `mysql_adaptor` plugin may not properly work when desired tables are missing and cannot deliver data.

abort No automatic schema upgrades are performed. The initialization is aborted and the program terminates. Schema upgrades can be done manually with `gromox-dbop(8gx)`.

Default: *skip*

64.4 See also

`gromox(7)`, `authmgr(4gx)`

PAM_GROMOX

65.1 Name

`pam_gromox` — a PAM plugin to authenticate with Gromox

65.2 Description

This module forwards authentication requests to the Gromox service plugins which themselves may pick MySQL or LDAP as a backend. `pam_gromox` is meant to be used in conjunction with non-Gromox SMTP/IMAP/etc. server processes that may be logically located before Gromox services and serving as accelerators.

65.3 Incantation in `/etc/pam.d/smtp`

Gromox accounts are not mapped from or to any Unix accounts, so the `pam_unix.so` module that is present in the default `/etc/pam.d/smtp` module list within Linux distributions is not suitable and can be wholly replaced. In otherwords, `/etc/pam.d/smtp` need just contain:

```
auth required pam_gromox.so
account required pam_permit.so
```

(`pam_gromox` does not provide a usable “account” handler, therefore “account required `pam_gromox.so`” would do nothing. The PAM framework always starts out with an initial deny policy, so at least one module needs to be called to make the PAM request succeed. For this reason, if there are no other “account” modules listed, `pam_permit.so` should be used.)

65.4 Configuration directives in `/etc/gromox/pam.cfg`

config_file_path

Colon-separated list of directories in which further configuration files, especially those used by plugin instances, will be searched.

Default: `/etc/gromox/pam:/etc/gromox`

pam_prompt

If `pam_gromox` detects the absence of a password but presence of a PAM conversation function, it will attempt to retrieve the password that way, and in doing so, will show this label just ahead of the nonechoing password prompt.

Default: `Password:`

service_plugin_ignore_errors

If set to yes, service plugins that fail to load on startup are ignored. If set to no, the daemon will exit if any plugin cannot be loaded.

Default: *no*

service_plugin_list

Path to a text file which lists the filenames of service plugins to load, one per line.

Default: (unspecified)

service_plugin_path

Path to a secondary directory where service plugins will be loaded from if a primary search in standard directories (as per ld.so(8)) was unsuccessful.

Default: */usr/lib/gromox*

65.5 See also

gromox(7)

66.1 Name

pop3 — Gromox POP3 server

66.2 Synopsis

pop3 [-c *config*]

66.3 Options

-c *config* Read configuration directives from the given file. If this option is not specified, */etc/gromox/pop3.cfg* will be read if it exists.

-version Output version information and exit.

-? Display option summary.

66.4 Configuration file directives

block_interval_auths

The amount of time a user is blocked from connecting to the service after too many failed logins.

Default: *1 minute*

cdn_cache_path Default: */var/lib/gromox/cdn*

config_file_path

Colon-separated list of directories in which further configuration files, especially those used by plugin instances, will be searched.

Default: */etc/gromox/pop3:/etc/gromox*

console_server_ipFP

An IPv6 address (or v4-mapped address) to expose the management console frontend on.

Default: *::1*

console_server_port

The TCP port number to expose the management console frontend on.

Default: *7788*

context_average_mem Default: *256K*

context_average_units

Lower clamp is 256.

Default: *1024*

context_max_mem Default: *2M*

context_num Default: *200*

data_file_path

Colon-separated list of directories in which static data files will be searched.

Default: */usr/share/gromox/pop3*

default_domain Default: (system domainname)

enable_capa_implementation

When enabled, the server will include an “IMPLEMENTATION” line in the CAPA response (RFC 2449 §6.9). This is disabled by default, as it can facilitate potential attackers’ information gathering.

Default: *no*

host_id Default: (system hostname)

listen_port

The TCP port to expose the POP3 protocol service on. (The IP address is fixed to the wildcard address.)

Default: *110*

listen_ssl_port

The TCP port to expose implicit-TLS POP3 protocol service (POP3S) on. (The IP address is fixed to the wildcard address.)

Default: (unset)

pop3_auth_times

The number of login tries a user is allowed before the account is blocked.

Default: *3*

pop3_certificate_passwd

The password to unlock TLS certificates.

Default: (unset)

pop3_certificate_path

Filesystem path to a certificate file for use with encrypted connection. The complete certificate chain should be present (as there is no other config directive to pull CA certs in).

Default: (unset)

pop3_conn_timeout

If a POP3 connection stalls for the given period, the connection is terminated.

Default: *3 minutes*

pop3_force_stls

This flag controls whether clients must utilize TLS, either by way of implicit TLS (cf. **listen_ssl_port**), or through the **STLS** command.

Default: *false*

pop3_private_key_path

Filesystem path to the key file needed to unlock the TLS certificate.

Default: (unset)

pop3_support_stls

This flag controls the offering of the STARTTLS extension/STLS command (RFC 2595) to clients.

Default: *false*

running_identity

An unprivileged user account to switch the process to after startup. To inhibit the switch, assign the empty value.

Default: *gromox*

service_plugin_ignore_errors

If set to yes, service plugins that fail to load on startup are ignored. If set to no, the daemon will exit if any plugin cannot be loaded.

Default: *no*

service_plugin_list

Path to a text file which lists the filenames of service plugins to load, one per line.

Default: (unspecified)

service_plugin_path

Path to a secondary directory where service plugins will be loaded from if a primary search in standard directories (as per ld.so(8)) was unsuccessful.

Default: */usr/lib/gromox*

state_path

Directory for runtime variadic data.

Default: */var/lib/gromox*

thread_charge_num

The maximum number of connections that each thread is allowed to process.

Default: *40*

thread_init_num

The minimum number of client processing threads to keep around.

Default: *1*

66.5 Files

- *data_file_path/pop3_code.txt*: Mapping from internal POP3 error codes to textual descriptions.
- */usr/lib/gromox/libgxs_*.so*: service plugins

66.6 See also

gromox(7), **midb_agent(4gx)**

67.1 Name

rtf2html — C++ helper program for the `mapi(4gx)` `mapi_decompressrtf` function

67.2 Description

`rtf2html` decodes an RTFCP file from standard input and converts the included RTF text to HTML, which is emitted on standard output. This utility does not support headerless RTF as emitted by word processors such as LibreOffice.

67.3 Options

`-version` Output version information and exit.

`-?` Display option summary.

67.4 Files

- `/usr/share/gromox/cpid.txt`: codepage number <-> name definition table

67.5 Normative references

- MS-OXRTFCP: Rich Text Format (RTF) Compression Algorithm

67.6 See also

`gromox(7)`

68.1 Name

sa.cfg — configuration file for `mkmidb`, `mkprivate`, `mkpublic`

68.2 Directives

backup_hash_num Default: *10*

backup_valid_days Default: *30*

config_file_path

Colon-separated list of directories in which further configuration files will be searched.

Default: */etc/gromox*

data_file_path

Colon-separated list of directories in which static data files will be searched.

Default: */usr/share/gromox*

extpasswd_type

A value ≥ 2 and ≤ 5 .

Default: *2*

gateway_mount_path Default: */var/lib/gromox/gateway*

http_accept_language

Ignore whatever Accept-Language was sent by HTTP clients and assume this new value.

Default: (unset)

log_file_path

Default: */var/log/gromox/sa.log*

logo_link

The hyperlink that is associated with the logo image.

Default: (unspecified)

mysql_dbname Default: *email*

mysql_host

The hostname/IP address for contacting the SQL server.

Default: *localhost*

mysql_password Default: (unset)

mysql_port

The TCP port number for contacting the SQL server.

Default: *3306*

mysql_username Default: *root*

oversea_relay_switch

Default: *true*

public_store_ratio Default: *10*

token_file_path Default: */run/gromox/sa*

ui_timeout Default: *10 minutes*

68.3 See also

gromox(7), mkmidb(8gx), mkprivate(8gx), mkpublic(8gx)

69.1 Name

`str_table` — string table match service plugin

69.2 Description

`str_table` implements checking for a trivial presence lookup of something in a text file. The plugin can be used in any Gromox process, but in practice only makes an appearance in `delivery-queue(8gx)` and `delivery(8gx)`.

The `str_table` plugin has multiple instances. These are: **`domain_list`**.

These instances are factual copies of the module to please the plugin loader and make it load different `.cfg` files. (This is clearly inefficient and may be fixed in a future release.) For example, the plugin loader will derive the name “`domain_list.cfg`” from the module name “`libgxs_domain_list.so`”, and then pass this filename to the module so it can be loaded as a config file.

The plugin supports temporary modification of the string table as it exists in memory through the telnet console mechanism. A reload of the table from disk purges these modifications.

69.3 Configuration file directives

`add_service_name`

The plugin instance will expose its function through this service function name. You should not generally this, because other plugins rely on certain fixed values.

Default: *instance_name_add*

`growing_num`

The maximum number of additional string table entries on top of what the on-disk table file has supplied. So when the table file contains 40 entries and `growing_num` is 100, the in-memory copy of the table can hold up to 140 entries total.

Default: *100*

`is_case_sensitive`

Flag which determines whether matches should treat uppercase and lowercase differently or not.

Default: *false*

`query_service_name`

The plugin instance will expose its function through this service function name. You should not generally this, because other plugins rely on certain fixed values.

Default: *instance_name_query*

remove_service_name

The plugin instance will expose its function through this service function name. You should not generally this, because other plugins rely on certain fixed values.

Default: *instance_name_remove*

69.4 Files

- *config_file_path/instance.cfg*: configuration file for the instance of *str_table* (e.g. */etc/gromox/delivery/domain_list.cfg* when *delivery(8gx)* was made to load *libgxs_domain_list.so*.)
- *state_path/instance.txt*, *config_file_path/instance.txt*: string table on which matches are carried out (e.g. */var/lib/gromox/domain_list.txt*)

69.5 See also

gromox(7)

70.1 Name

textmaps — Service plugin for various data

70.2 Description

textmaps is a service plugin that reads various data maps into memory and offers lookups in them.

It is vital that it be able to load the text files and initialize the mappings, as otherwise, character set conversions cannot be performed and e.g. `exchange_nsp(4gx)` responds with failure to RPCs made by Outlook that involve character set IDs or locale IDs.

70.3 Files

- `data_file_path/cpid.txt`: mapping between character set IDs and names
- `data_file_path/lang_charset.txt`: mapping from language code to character set
- `data_file_path/lcid.txt`: mapping between locale IDs and names
- `data_file_path/mime_extension.txt`: mapping between file extensions and MIME types

`data_file_path` is determined by the configuration of the program that loaded the textmapplug plugin.

70.4 See also

`gromox(7)`

71.1 Name

timer — deferred command executor

71.2 Synopsis

timer [-c *config*]

71.3 Description

The timer daemon can be used to schedule commands to be executed once, at a particular time in the future. It is similar to the at(1) command and its associated daemon, atd.

timer(8gx) generally receives commands from timer_agent(4gx).

71.4 Options

-c *config* Read configuration directives from the given file. If this option is not specified, /etc/gromox/timer.cfg will be read if it exists.

-version Output version information and exit.

-? Display option summary.

71.5 Files

- *config_file_path*/timer_acl.txt: A file with one address (IPv6 or v4-mapped) per line of allowed clients. In its absence, ::1 is default-whitelisted.
- /var/lib/gromox/timer.txt: This file is used to save the state of timer(8gx) and persist them across restarts.

71.6 Configuration file directives

config_file_path

Colon-separated list of directories which will be scanned when locating further configuration files, especially those used by plugin instances.

Default: */etc/gromox/timer:/etc/gromox*

running_identity

An unprivileged user account to switch the process to after startup. To inhibit the switch, assign the empty value.

Default: *gromox*

timer_listen_ip

An IPv6 address (or v4-mapped address) for exposing the timer service on.

Default: *::1*

timer_listen_port

The TCP port number for exposing the timer service on.

Default: *6666*

timer_state_path Default: */var/lib/gromox/timer.txt*

timer_threads_num

The minimum number of client processing threads to keep around.

Default: *50*

71.7 Timer protocol

The timer service is exposed as a line-based text protocol. Upon connection, the event server gratuitously writes “OK”, following which the server will wait for timer commands, and execute them synchronously.

The command “ADD <seconds> <command>” installs a new timer for the given command to be executed in that many seconds from now. The server will respond with “FALSE 2”, “FALSE 3”, or respond with the timer ID as “TRUE <id>”.

The command “CANCEL <id>” revokes the timer with the chosen ID.

71.8 See also

gromox(7), **timer_agent(4gx)**

72.1 Name

`timer_agent` — Service plugin for deferred command execution with `timer(8gx)`

72.2 Description

`timer_agent` connects to a remote `timer(8gx)` daemon and locally installs two service functions, “`add_timer`” and “`cancel_timer`”, which can be used to set up and rescind, respectively, jobs for later execution.

In practice, this is used by `exchange_emsmb(4gx)` and `zcore(8gx)` to implement delayed sending of messages.

72.3 Configuration file directives

connection_num

Number of connections to keep active.

Default: 8

timer_host

The hostname/IP address for contacting the timer daemon.

Default: `::1`

timer_port

The TCP port number for contacting the timer daemon.

Default: 6666

72.4 See also

`gromox(7)`, `timer(8gx)`, `at_client(4gx)`

73.1 Name

`user_filter` — Service plugin for application of user login limits

73.2 Configuration file directives

`add_service_name`

The plugin instance will expose its function through this service function name. You should not generally this, because other plugins rely on certain fixed values.

Default: `instance_name_add` (e.g. `user_filter_add` for `libgxs_user_filter.so`)

`audit_interval` Default: `1minute`

`audit_max_num` Default: `0`

`audit_times` Default: `10`

`grey_growing_num` Default: `0`

`growing_num`

The maximum number of additional string table entries on top of what the on-disk table file has supplied. So when the table file contains 40 entries and `growing_num` is 100, the in-memory copy of the table can hold up to 140 entries total.

Default: `0`

`is_case_sensitive`

Flag which determines whether matches should treat uppercase and lowercase differently or not.

Default: `false`

`judge_service_name`

The plugin instance will expose its function through this service function name. You should not generally this, because other plugins rely on certain fixed values.

Default: `instance_name_judge`

`query_service_name`

The plugin instance will expose its function through this service function name. You should not generally this, because other plugins rely on certain fixed values.

Default: `instance_name_query`

`temp_list_size` Default: `2000`

73.3 Files

- *config_file_path/instance.cfg*: configuration file for the instance of `str_filter` (e.g. `/etc/gromox/http/user_list.cfg` when `http(8gx)` was made to load `libgxs_user_list.so.`)
- *state_file_path/instance.txt*: greylist

73.4 See also

gromox(7)

74.1 Name

zcore — Bridge for PHP-MAPI requests

74.2 Synopsis

zcore [-c *config*]

74.3 Description

zcore is a bridge process (proxy) between mapi(4gx) and exmdb_provider(4gx). It listens on /run/gromox/zcore.sock (hardcoded) for zcore RPCs, a Gromox-specific protocol and issues exmdb RPCs to exmdb_provider(4gx). As exmdb_provider connections have no state to speak of, zcore is the process that defines the logins sessions. zcore needs to run on the same server as the program that uses the Z-MAPI PHP functions.

74.4 Options

- c *config* Read configuration directives from the given file. If this option is not specified, /etc/gromox/zcore.cfg will be read if it exists.
- version Output version information and exit.
- ? Display option summary.

74.5 Network protocol

The transmissions on the zcore socket are simple concatenations of protocol data units built using the NDR format. The PDU length is present within the PDU itself near the start.

```
{
    leuint32_t length;
    char pdu[];
}
```

```
pdu := {
    uint8_t call_id;
    string directory;
    switch (call_id) {
        ...
    }
}
```

74.6 Store lookup

zcore determines the store path for a user from the user database, which may be provided by a service plugin like `mysql_adaptor(4gx)`.

The filemap that specifies how paths are handled is located at `data_file_path/exmdb_list.txt`, whereby `data_file_path` is the eponymous directive from the config file.

Each line in this file consists of 4 columns separated by whitespace:

- A portion of the store path to match on
- The type of store (“private” or “public”)
- An IPv6 address (or v4-mapped address) of the server running `exmdb_provider(4gx)`
- The TCP port number of the server

74.7 Files

- `data_file_path/exmdb_list.txt`: exmdb multiserver map
- `data_file_path/folder_lang.txt`: Translations for essential folders in a message store.
- `data_file_path/langmap.txt`: Mapping between translation names and libc locale names.
- `data_file_path/msgchg_grouping/`: Data files for `exchange_emsmb(4gx)`
- `data_file_path/notify_bounce/`: A directory with translation files for various bounce notifications.
- `/usr/lib/gromox/libgxs_*.so`: service plugins

74.8 Configuration file directives

address_cache_internal Default: *5 minutes*

address_item_num Default: *100000*

address_table_size Default: *3000*

config_file_path

Colon-separated list of directories which will be scanned when locating further configuration files, especially those used by plugin instances.

Default: `/etc/gromox/zcore:/etc/gromox`

console_server_ip

An IPv6 address (or v4-mapped address) to expose the management console frontend on.

Default: `::1`

console_server_port

The TCP port number to expose the management console frontend on.

Default: *3344*

data_file_path

Colon-separated list of directories which will be scanned when locating data files.

Default: */usr/share/gromox/zcore*

default_charset Default: *windows-1252*

default_timezone Default: (unspecified)

freebusy_tool_path Default: */usr/libexec/gromox/freebusy*

host_id Default: (system hostname)

mailbox_ping_interval Default: *5 minutes*

mail_max_length Default: *64M*

max_ext_rule_length Default: *510K*

max_mail_num Default: *1000000*

max_rcpt_num

The maximum number of recipients that an e-mail is allowed to have.

Default: *256*

notify_stub_threads_num Default: *10*

rpc_proxy_connection_num Default: *10*

separator_for_bounce Default: *;*

service_plugin_ignore_errors

If set to yes, service plugins that fail to load on startup are ignored. If set to no, the daemon will exit if any plugin cannot be loaded.

Default: *no*

service_plugin_list

Path to a text file which lists the filenames of service plugins to load, one per line.

Default: (unspecified)

service_plugin_path

Path to a secondary directory where service plugins will be loaded from if a primary search in standard directories (as per `ld.so(8)`) was unsuccessful.

Default: */usr/lib/gromox*

smtp_server_ip

SMTP server hostname or address to contact for outgoing mail.

Default: *::1*

smtp_server_port

SMTP server TCP port number to contact for outgoing mail.

Default: *25*

state_path

Directory for runtime variadic data.

Default: */var/lib/gromox*

submit_command Default: */usr/bin/php /usr/share/gromox/sa/submit.php*

user_cache_interval

Sets the time how long a zcore-specific “OBJECT_TREE” structure is cached. At the end of this interval, user settings like PR_EC_WEBACCESS_JSON (stored in a separate file, not in SQLite) are flushed to disk.

Default: *1 hour*

user_table_size Default: *5000*

x500_org_name Default: (unspecified)

zarafa_mime_number Default: *4096*

zarafa_threads_num

The minimum number of client processing threads to keep around.

Default: *100*

zcore_listen

The named path for the AF_LOCAL socket that zcore will listen on.

Default: */run/gromox/zcore.sock*

zrpc_debug

Log every incoming zcore RPC and the return code of the operation in a minimal fashion to stdout. Level 1 emits RPCs with a failure return code, level 2 emits all RPCs.

Default: *0*

74.9 Files

- *config_file_path/exmdb_list.txt*: *exmdb multiserver selection map, see exmdb_provider(4gx) for details.*
- *data_file_path/langmap.txt*: *mapping between language code and its corresponding libc locale name.*
- *data_file_path/msgchg_grouping/.txt**
- *data_file_path/notify_bounce/*: *response templates for when mail cannot be delivered*
- */var/lib/gromox/user/*/config/zarafa.dat*: *file for the* PR_EC_WEBACCESS_JSON property.*

74.10 Notes

Behavior for the address book generally mirrors *exchange_nsp(4gx)*, so see that manpage for additional notes.

74.11 See also

gromox(7)

LEGAL NOTICE

Amazon Web Services, AWS, AWS CloudFront, CloudFront, Elastic Compute Cloud, Amazon EC2, EC2, Elastic Load Balancer, ELB, and related graphics, logos, page headers, button icons, scripts, and service names are trademarks, registered trademarks, or trade dress of Amazon Web Services in the U.S. and/or other countries.

Ansible, CentOS, CoreOS, RHEL, Red Hat Enterprise Linux, and RHCE are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

Apache and the Apache feather logo are registered trademarks of The Apache Software Foundation in the United States and other countries.

Azure, Azure Container Service, ACS, Azure Resource Manager, Microsoft Windows Server, Active Directory, and Microsoft are registered trademarks or trademarks of the Microsoft Corporation.

Docker is a registered trademark of Docker, Inc.

ElasticSearch, Logstash, and Kibana are registered trademarks of Elasticsearch B.V., registered in the U.S. and other countries.

Google is a registered trademark, and Google Cloud Platform, GCP, Google Cloud Platform service, GCP infrastructure platform, Google Cloud enterprise services, and TensorFlow open-source software library are trademarks of Google, Inc.

Grafana is a trademark of Coding Instinct AB.

HAProxy is a registered trademark in the United States and France.

JavaScript, MySQL, and Oracle are trademarks or registered trademarks of the Oracle Corporation and/or its affiliates.

Debian is a registered trademark of SOFTWARE IN THE PUBLIC INTEREST, INC.

Kubernetes is a graduated project of the Cloud Native Computing Foundation, or CNCF. Cloud Native Computing Foundation, CNCF, Kubernetes, K8s, Prometheus, and The Linux Foundation are registered trademarks of the The Linux Foundation. CNI and Spinnaker are trademarks of The Linux Foundation. Additional trademarks and registered trademarks can be found at <https://www.linuxfoundation.org/trademark-list/>.

Mac, MacBook, and macOS are registered trademarks of Apple, Inc in the United States and other countries.

Nagios, the Nagios logo, and Nagios graphics are the service marks, trademarks, or registered trademarks of Nagios Enterprises.

nginx is a trademark of Nginx Software, Inc.

OpenID Connect and OpenID are registered trademarks of the Open ID Foundation in numerous countries.

Percona, PerconaXtraDB, PerconaXtraDB Cluster, Percona Server, XtraDB, are registered trademarks or trademarks of Percona.

Splunk is a registered trademark of Splunk, Inc. in the United States and other countries.

Telegraf is a trademark of InfluxData, Inc.

Ubuntu and Canonical are registered trademarks of Canonical Ltd.

UNIX is registered trademark of The Open Group in the United States and other countries.

VMware ESXi and VMware are registered trademarks of VMware, Inc. in the United States and other countries.

All other trademarks, trade names, service marks, and companies referenced herein belong to their respective companies, foundations, or development communities.

Copyright © 2020-2021 grommunio GmbH