

```

-----+-----
|                                     |
|               TomeNET Administrator Guide               |
|-----+-----|
| Latest update: 24. May 2026 - written by C. Blue (c_blue@gmx.net) |
| for TomeNET version v4.9.4 - official websites are:           |
| https://www.tomenet.eu/ (official main site, formerly www.tomenet.net) |
| https://muuttuja.org/tomenet/ (Mikael's TomeNET site)         |
|
| This guide is meant for TomeNET server administrators, not for players.
| The guide for players is just called "TomeNET Guide" (without 'Admin').
|
| If your text editor cannot display the guide properly (needs fixed-width
| font like for example Courier), simply open it in any web browser instead.
+---+
| Welcome to this guide!
| Although I'm trying, I give no guarantee that this guide is always correct.
| Don't blame me if something differs or is missing; it shouldn't though.
|
| If you have any suggestions about the guide or the game, please use the
| /rfe command in the game or write to the official forum on www.tomenet.eu.
|
|
\

```

Contents

```

-----
(0) Administrative slash commands - 1
    (0.1) Shutting down the server gracefully - 2
    (0.2) Live LUA changes - 2
    (0.3) More commands - 2
(1) Accounts
    (1.1) The account editor 'accredit'
    (1.2) New accounts and validation
    (1.3) The administrator account
        (1.3a) Dungeon Master
        (1.3b) Dungeon Wizards
    (1.4) Commands to change powers of individual accounts
(2) Admininstrator menus
    (2.1) The administrator menu
    (2.2) The knowledge menu
    (2.3) Other admin-enhanced menus and keys
    (2.4) Interaction leeway
(3) Server command-line options
(4) Server configuration files
(5) Compilation
    (5.1) Server compilation
        (5.1a) Server recompilation
        (5.2) accredit compilation

```

```

=====
Issuing slash commands is done in-game, by pressing ':' key to open the chat
prompt and entering the command there, starting with the slash '/' symbol.
This guide only mentions slash commands that are specifically for use by
administrators or players with elevated privileges and thereby not accessible
to normal player accounts.

```

Account-level related commands (validation, privileging) are listed in (1.4).

All admin-specific slash commands can be found in file 'src/server/slash.c' starting after the code "else if (admin) {".

Note that unlike player-usable slash commands, admin slash commands sometimes don't have guard rails so it might be easy to crash the server if you specify invalid arguments to a command (resulting in a 'Panic Save' log message, which means the server realized it's going to crash and quickly saves all active

players and the world state to savefiles on the disk to preserve the current progress).

(0.1) Shutting down the server gracefully

Log in with a Dungeon Master or Dungeon Wizard and issue the slash command
/shutdown 0

This will initiate an orderly shutdown with proper cleanup and therefore is much preferable over things such as spamming CTRL+C into the server's terminal or to kill it otherwise.

To make the server wait for all players to leave the dungeons before executing the shutdown, use this command:

/shutempty

To cancel a pending '/shutempty' command again, use:

/shutcancel

(0.2) Live LUA changes

To reinitialize the LUA framework immediately without the need for a server restart, use this command:

/initlua

Use this if you changed any LUA files and want to put the changes live immediately, mainly for the file 'lib/scpt/custom.lua' (compare (4)).

(0.3) More slash commands

/ ... (<- note the space that must follow after the slash!)
/script ...
/lua ... These commands all do the same: Execute a LUA script command on the server. Example: '/ prn("Hello, world!")'
 This can also be done by entering the admin menu with ']', then (6) System, 'e'.
 Also see (0.4) about issuing LUA commands.

// ... (<- note the space that must follow after the slashes!)
 This command executes a LUA script command on the client.
 This can also be done by entering the admin menu with ']', then (6) System, 'c'.
 This is not really an administrative command, as it is run purely on the client-side, so anyone can actually use it.
 Can be helpful for debugging purpose.

/log View the server's log file 'lib/data/tomenet.log', ie the same that is shown in the terminal where the server is running.
 Can also be viewed by entering the admin menu with ']', and then -> (6) System Commands -> (1) View tomenet.log.

/rec The recall command works almost-instantly for admins and has no range limit. Admins also do not need any recall items/spells, they can use this command freely, at any time.
 They can also use an extra syntax, adding '/x,y' for a specific grid in the target worldmap sector, eg:
 /rec 32,32/15,15

/ex The extra-info command shows additional info for admins.

/wish <tval> <sval> [<xNum>] [<bpval> <pval> <name1> <name2> <name2b>]
(DEPRECATED) Creates an item and puts it into your inventory.
 This command's syntax is rather complex and yet the /wish

command is lacking certain features. For that reason I do not recommend using this function anymore; I created three simpler, yet also more powerful versions, these are:

```
/nwish [[<#skip>:][#amount ][\<prefix-ego>\<item name>[\<postfix-ego>]
      [+<bpval>][^<pval>]
      Creates an item and puts it into your inventory.

Common usage examples for XP/stat potions:
/nwish ex      -- Potions of Experience
/nwish str     -- Potion of Strength, likewise for 'int',
               'wis', 'dex'.
/nwish 4:con   -- Potion of Constitution (as there are 4
               other items having 'con' in their name
               first, in the 'lib/game/k_info.txt' file).
               Or instead, for a direct match:
/nwish cons
/nwish 17:cha  -- Potion of Charisma, lots of CHA in mail
               and stuff before it >:) - or:
/nwish 1:chari -- still have to skip 'Amulet of Charisma'

Or for multiple items:
/nwish 1:3 chari -- creates 3 Potions of Charisma.

/awish [<#skip>:]<artifact name>
      Creates a static artifact (true artifact) and puts it into your
      inventory.

/swish [<#skip>:][#amount ]<spell name>
      Creates a specific spell scroll and puts it into your
      inventory.

/qinf      Prints information about all available quests. These are
           defined in 'lib/game/q_info.txt' and optionally additionally in
           'lib/scpt/quests.lua'.

/qstart    Starts (enables) a quest that is currently inactive.

/qstop     Stops (disables) a quest that is currently active.
           Chaining the two commands, first '/qstop' then '/qstart' will
           therefore reset a quest, in case questors disappeared etc.
```

(0.4) LUA commands

The slash command "/" (note the space after the slash) can be used to execute arbitrary LUA code on the server (as mentioned above in (0.3)).
The LUA scripting language in TomeNET is the old 4.0.1 version, so attempting to use commands of newer LUA versions may fail.

Whenever there is no specific admin slash command to do something, chances are you can just use LUA code to achieve your goal.
All functions and variables you can access in LUA are defined in these files, so you can browse those to see if there's anything you could use:

```
lib/scpt/*.lua      -- the scpt folder contains all the LUA script files.
src/server/lua_bind.c -- a bunch of functions created for use in LUA.
src/server/player.pre
src/server/spells.pre
src/server/util.pre  -- these three files declare functions and variables
                     of the TomeNET C code to be also available in LUA.
```

Also important here is the file 'src/common/types.h' which contains for example the 'player_type' which is the structure of the 'player'/'players' variables in LUA to access live player data (ie currently logged on characters).
Looking there you can for example see that the 'struct player_type' has -among

many others- the properties 'pclass', 'prace', 'ptrait' and 'mode'.
With this knowledge you can manipulate online characters:

Examples of an admin char using LUA scripting:

```
/ player.mode=8      -- sets the character mode of this character to 8,  
                    which corresponds to MODE_EVERLASTING (compare file  
                    'src/common/defines.h').  
/ players("Gandalf").pclass=1  -- sets the class of the currently logged  
                               in character 'Gandalf' to 1, which  
                               corresponds to CLASS_MAGE aka 'Istar'  
                               (compare file 'src/common/defines.h'  
                               again).
```

Note that some LUA changes might not be 'refreshed' instantly in all aspects of
the live game code in RAM, so sometimes it might be required to for example:

- press CTRL+R to redraw the windows,
- relog (log out + log in again) the affected character,

in order to see the changes.

There is a quirk in LUA: Arrays start at index 1, not 0 (as in C).
So if you manipulate array elements you might need to add +1 to the index.

(1) Accounts

=====

Everyone who interacts with the server within the game world uses an 'account',
defined by login name and login password.

(1.1) The account editor 'accredit'

Accounts can be moderated and managed with the account editor 'accredit', which
resides in 'src/account' folder. The compiled 'accredit' binary must be placed
into the same folder where file 'tomenet.acc' resides, usually 'lib/save'.
This tool can be used to validate, invalidate or change privileges of players,
and much more. However, it is usually simpler and more comfortable to use
administrative slash commands instead, issued in-game on an administrator
character, ie a DM or DW (see below), see (1.4).

For compilation of accredit see (5.2) much further down.

(1.2) New accounts and validation

Whenever someone logs in to the TomeNET game server for the first time ie
with an account name that is not yet known to the server, the server
automatically creates a new account for this player.

These accounts start out as 'invalid', which means they are restricted in
what they can do and how far they can level up, until they are 'validated'
by a server administrator character (by using '/val' command, see (1.4))
or via the 'accredit' program.

Only the first account ever created on a server is not automatically an
invalid player account, instead it will become...

(1.3) The administrator account

The very first account created on a server will automatically become the
main administrative account: The "dungeon master" (DM) account.

All characters on an administrative account will start out with different
items than players, none of which will be equipped automatically:

- a Handbook of Dungeon Keeping
- 3 Rods of Probing
- an Administrative Decree -- aka amulet of *Invincibility*. This will always prevent any kind of harm and at the same time will make all monsters ignore you (note that DMs are already ignored by all monsters anyway).
- an Amulet of *Invulnerability* -- this is simimilar to the decree but will not prevent monsters from targetting you if you are a Dungeon Wizard. Useful for testing stuff.
- an Amulet of *Immortality* -- same as the Invulnerability amulet but you will take damage, you just won't die from it.
- a Dwarven Lantern of Brightness
- The Shadow Cloak of the Dungeon Master
- The Goggles of the Dungeon Master -- This allows a DW (or any char) to actually see unseeable DM chars.
- The Scythe of the Dungeon Master -- This deals monster's HP + 1 to any monster, thereby one-shotting them, except if they have NO_DEATH flag. Inscribe it 'IDDQD' to even kill monsters that have NO_DEATH flag. If this weapon is worn, the DM will not auto-switch places with monsters when running/bumping into them, but instead stop when running and attack when bumping. If the scythe is inscribed '@Ox' the DM will still auto-switch places with monsters on running into them.
- A Long Bow of Power and a Magic Arrow to go with it.

(1.3a) Dungeon Master

The character of the same name as an administrative account will be the Dungeon Master (DM) character, which means it'll automatically be a ghost while still retaining the powers of carrying items and doing basically everything a living character could.

At the same time it can move through any kind of walls except for level boundaries and can also do various things that normal player characters cannot do, both in terms of gameplay interactions and also in usage of specific slash commands that are not available to normal players.

The dungeon master moves at +50 base speed and always can see all players and monsters telepathically, even those without a mind. The dungeon master is level 100, while all other chars stop at 99. This has no particular effect. The dungeon master is always ignored by all monsters as if they weren't there.

The DM has particular movement behaviour, mentioned above in (1.3) under the item explanation for 'The Scythe of the Dungeon Master'.

Last but not least the DM character is always invisible to everyone (except for other DM characters) and does not appear on the players list ('@' key), unless this is changed explicitly in the tomenet.cfg file.

(1.3b) Dungeon Wizards

All other characters on the DM account will become Dungeon Wizards (DW), which are not ghosts but still have full administrative capabilities and therefore can use all administrative slash commands same as the Dungeon Master can. The DWs are fully visible to everyone, just as any other character.

(1.4) Commands to change powers of individual accounts

Instead of modifying account status using the 'accredit' tool, it's usually much easier and more comfortable to log in with a Dungeon Master or Dungeon Wizard and use administrative slash commands to moderate player accounts and change their privileges:

Privileged commands

These commands can be used by DMs, DWs but also by normal players whose accounts were set to 'privileged mode' by an administrator.

/val <name> Validate a newly created player account, so it can be used for normal gameplay without 'trial mode' restrictions, such as:
 - no levelling beyond 25.
 - no private chatting.
 - no dropping items.

/inval <name> Invalidate a player account, reapplying 'trial mode' restrictions. This is for example done on the official servers where a strict one-account-per-player rule is enforced, when an admin notices that a player uses more than one account.

/linv List all newly created invalid accounts (so you can decide to validate them one by one using the /val command then.)

Administrative commands

These commands can only be used by administrators (DMs and DWs).

/privilege Set a player account to 'privileged' mode, unlocking the 3 commands listed above. Sometimes, some quests in 'testing' stage might be accessible to privileged players for early testing.

/vprivilege Set a player account to 'very privileged' mode.
 Has no further benefits over privileged mode, but may sometimes allow early testing of new features or commands if available.

/unprivilege Set a player's account back to non-privileged aka normal.

/makeadmin Turns a player account into an administrative account (see (1.3) for administrative accounts). All its characters will turn into DM (if same name as account) and DWs. Use with care as this action cannot be undone; instead, you will have to invoke the 'accredit' tool and 'D'eleate the account there, to get rid of it again.
 Alternatively to 'accredit' you can also use the admin menu in-game, invoked with ']' key ->Player Commands ->Delete Player which asks for an account name and will delete all characters on that account and the account itself (compare (2.1)).

When logged in on an account you can check its status by pressing = A.

/vinv <index> Validate a specific entry of the list shown by '/linv' command.

/dinv <index> Discard a specific entry of the list shown by '/linv' command. (The account will remain invalid, it's just no longer shown on the 'newly created accounts' list. You can still validate it by using the '/val' command anytime.)

/clinv Clears all entries from the list shown by '/linv' command.

(2) Administrator menus

=====

(2.1) The administrator menu

When logged on as DM or DW you can press the ']' key in-game to invoke the administrator menu:

- (1) Level Commands
 - (3) Add dungeon -- use this to place an IDDC dungeon on a newly created server; make sure its coordinates match the ones the server compiled with in 'src/common/defines-local.h'.
The standard IDDC flags to enter here are:
"Theme" 0; "Base level" 1; "Max depth" 127;
"Is it a tower?" - check defines-local.h:
If you left WPOS_IRONDEEPDIVE_Z to -1 then no, if you set it to 1 then yes;
"Generate towns inbetween?" y,
"Generate towns randomly?" n,
"Disallow simple stores?" y,
"Generate hidden library?" y,
"Custom DF1/2/3 flags" 0.
- (2) Building Commands
- (3) Summoning Commands
- (4) Generation Commands
- (5) Player Commands
 - (6) Delete player -- fully deletes an account and all its chars.
- (6) System Commands
 - (1) View tomenet.log -- same as invoking '/log' command
 - (e) Execute script command -- same as invoking '/' command
 - (c) Execute local script command -- same as invoking '// ' command

(2.2) The knowledge menu

Some submenus in the knowledge menu ('~') provide additional info for admins:

- (1) Artifacts found -- also shows artifact carrier and timeout
- (3) Unique monsters -- also shows 'unseen' uniques.
- (b) Other players' equipment -- (same as the ' key) for admins shows inventory+equipment of all players.
- (8) Recall depth and towns -- shows complete info for admins
- (9) Houses -- shows complete houses list, with extra info
- (0) Wilderness map -- shows the complete world map

(2.3) Other admin-enhanced menus and keys

The Mathom-house in Bree has a new submenu for admins:

- f) Static floor records -- shows all dungeon floors currently static, usually from some perma-/ghost-death, but also from someone having logged out there.

Keys:

- ' to view other players' inventory+equipment (compare (2.2), ~b).
- d admins can drop any items, always.
- g/5 to pick up an item always works for admins (even as ghost).
- h admins can modify any houses even if they are owned by others, and they can permanently DELETE a house with 'D' (don't do it).
- I admins can always inspect any item for full knowledge.
- k admins can destroy any item, including artifacts.
- t T admins can take off any items, always.
- @ admins can see additional info in the players list (location).

In shops:

- o Dungeon Masters will receive item orders immediately on revisiting the shop.

(2.4) Interaction leeway

Admins can often perform certain game commands, actions, interactions even though their character does not meet required criteria. Sometimes this only works only for DMs, in most cases it works for DWs too.

In most cases they will see the usual failure/error message that a normal player character who tries this would see, but for the admin the action they tried to perform will actually be accepted and executed successfully despite the error message!

This makes testing certain things much easier as you do not need to modify the admin character first in all necessary ways to meet the requirements for certain actions.

Example 1: Admins can send items via telekinesis to any character, even those of incompatible mode and even to ghosts. In the specific case of sending to a soloist-mode character they will still see an error message but nevertheless the spell will go through successfully. There is no specific rule when this works or not, and when there will still be a message or not, it's kinda random/arbitrary, as this was usually hot-coded for quick debug/prototype reasons, for the eyes of developers only. :-p

Example 2: Admins can tunnel even if they are ghosts.

(3) Server command-line options

=====
The server binary (usually 'tomenet.server') can be invoked with command-line options.

Some of these are only available the first time a new TomeNET server is started as they will affect the random world generation - a process that is only done the very first time a server starts.

If you want to change the server's world again with those particular options labeled 'On server creation', you must delete the file 'lib/save/server' which contains the world. This will reset the server's world to a fresh state.

All command-line options available are:

```
-h          Just display all available command-line options and exit.
-r          Reset the server (implies -w and -f)
-w          Reset the server partially: New wilderness
-f          Reset the server partially: New flavours
-a          (On server creation!) Ensure that all terrain types are created
-b          (On server creation!) Don't allow watery wilderness around Bree
-l          (On server creation!) Place low-level dungeons not far from Bree
-H          Reinitialize houses
-z          Don't catch signals
-u<path>   Look for user files in the directory <path> (UNUSED)
-s<path>   Look for save files in the directory <path>
-t<path>   Look for text files in the directory <path>
-m<file>   Specify configuration <file>
```

(4) Server configuration files

=====
Various files influence the server configuration and the world.

- lib/config/tomenet.cfg
This is the main configuration file, which the server reads each time it starts up.
- src/common/defines*.h
Header files starting on 'defines...' that contain a lot of server settings, most notably the world map location of the 'Ironman Deep Dive Challenge' dungeon (defines-local.h).
If anything in the header files is changed, a full, clean recompilation of the server is required (see (5.1a)).
- lib/scpt/custom.lua
LUA code in here can manage things such as starting scheduled events or season changes.
Any changes to LUA files can be put live instantly if a DM/DW issues the '/initlua' command, which reinitializes the LUA framework without the need for a server restart.

(5) Compilation

=====

Anything described here is for Linux.

Unlike the TomeNET client the server side has way less package requirements. See TomeNET-Guide.txt (2.1b) for basic requirements, only some of that will be needed for the server, eg having a compiler installed - default is 'clang' but you can edit 'src/makefile' to change 'CC = clang' to 'CC = gcc' if you prefer.

(5.1) Server compilation

To compile the server, assuming you are in the 'tomenet' folder, switch to folder 'src':

```
cd src
```

and issue clean-compilation commands:

```
make clean
make tomenet.server
```

Now move the binary up one level so it's no longer in 'src' but in the main tomenet folder:

```
mv tomenet.server ..
```

And then switch back up one folder to the binary:

```
cd ..
```

Now if it's the first time running this server, it will create the world. For this reason consider (3) for potential command-line options you might wish to add. Here are two examples,

- (a) for first-time-ever server startup that generates the randomized world, ie the main server savefile at 'lib/save/server', and...
- (b) for a normal subsequent startup as they happen all the time:

a) First time server-startup, to randomly generate the game world:

```
./tomenet.server -a -b -l
```

b) Just a normal, subsequent server-startup after the world was already created so we don't need to (and cannot) specify any world-creation-related options:

```
./tomenet.server
```

(5.1a) Server recompilation

If you modify any C files, especially the header files (*.h files, see (4) about defines*.h config files), use these same instructions to recompile the server as given above in (5.1). Assuming again that this isn't the very first server startup ever, ie the file 'lib/save/server' already exists, run the

recompiled server as in the example (b) without any world-creation-related command-line options:

```
./tomenet.server
```

(5.2) accedit compilation

To compile the 'accredit' account editor, assuming you are in the 'tomenet' folder, switch to folder 'src':

```
cd src
```

and type:

```
make accedit
```

Now move the resulting 'accredit' binary to the server's savegame folder:

```
mv accedit ../lib/save
```

and return to the tomenet folder:

```
cd ..
```

Then you can run 'accredit' by going into folder 'lib/save':

```
cd lib/save
```

and then invoking the binary:

```
./accredit
```

The reason we moved accedit to the save folder is that it must find the file 'tomenet.acc' in its folder, and tomenet.acc is located in 'lib/save'.

Alternatively you can just sym-link the tomenet.acc file to any location you prefer and put accedit there accordingly, for example the root 'tomenet' folder.

=====