

Nostrian Conquest

Sysop Manual

A Rust recreation inspired by the classic 1990s BBS door game Esterian Conquest.

Version 1.0.0-beta.5 — Beta

Revision date: October 25, 2026



This manual © 2026 Mason A. Green and is licensed under CC BY-NC-SA 4.0.

License text: <https://creativecommons.org/licenses/by-nc-sa/4.0/>

Contents

1. Introduction	2
2. Getting Started: Hosting a Campaign	2
2.1. System Requirements	2
2.2. Building from Source	2
2.3. Package Distribution	3
2.4. Choose Your Deployment	3
2.4.1. Localhost Session	3
2.4.2. BBS Door Host	3
2.5. Game Directory Layout	4
2.6. Initializing a New Game	4
3. Game Directory Structure	4
4. Configuration	5
4.1. Direct nc-game	5
4.2. BBS Door Host	5
4.3. AI Seats	5
4.4. Game settings: config.kdl	6
4.5. Display Defaults	7
5. Terminal Access	7
6. Localhost Session Setup	7
7. BBS Door Setup	7
7.1. Flags for Door Mode	8
7.2. Drop File	8
7.3. Verified BBS Hosts	10
8. File-Based Turn Submission	10
9. Turn Processing and Maintenance	11
9.1. Checking a Campaign	12
10. Player Management	12
10.1. Reserving Seats	13
10.2. Exporting a Player's Star Map	13
10.3. Checking a Player's Economy	14
11. Terminology	14
12. CLI Reference	14
12.1. nc-sysop	14
12.2. nc-game and nc-door	15

1. Introduction

Nostrian Conquest is a multi-player strategy game inspired by a DOS-era BBS classic. This Rust edition gives the sysop a direct player client, a BBS door, and the tools needed to run a campaign.

This manual is for the **sysop** — the man who hosts the game. It covers:

- choosing the live deployment path
- creating and maintaining a game directory
- running a local game
- putting `nc-door` on a BBS
- managing players and yearly maintenance

For player-facing rules and gameplay, see the **Player Manual**.

To reach the author with questions, bug reports or ideas, message this Nostr public key. The contact page also lists Discord and Matrix.

```
npub1cecl2g9z776jvpudpqxajrsrgwuvq4vp7w26q74tx2438zy3q9vq8st7pp
```

This manual is the authoritative sysop manual for the Rust edition. The preserved `.doc` files in `original/v1.5/` are historical reference. Use them as a fallback when classic operator intent is unclear.

2. Getting Started: Hosting a Campaign

2.1. System Requirements

- Linux, macOS, or Windows
- Rust toolchain (stable) for normal localhost builds
- A terminal emulator for localhost play
- A BBS server with socket support, only if you want legacy door deployment

2.2. Building from Source

From the repository root:

```
cd rust
cargo build --release -p nc-sysop -p nc-game
```

This builds all three binaries: `nc-sysop`, `nc-game`, and `nc-door`. The `nc-game` package holds both `nc-game` and `nc-door`, so `-p nc-game` builds both. The release bundle script runs the same build. The release binaries are in `rust/target/release/`.

IMPORTANT: If you update to a newer build and the game reports an unsupported runtime schema, the `ncgame.db` in that directory is stale for that build. Recreate or refresh `ncgame.db` before you launch again. `nc-door` shows this as an in-app error screen instead of dropping back to the shell.

2.3. Package Distribution

BBS sysops may use a packaged `nc-sysop` archive or build from source. Localhost sysops build from source and run `nc-game` directly.

Keep the binary roles straight.

`nc-game` is the direct localhost session client. `nc-door` is the BBS door entrypoint on Windows and Linux. `nc-sysop` is the administrator's tool for creation, settings, reservations, and maintenance.

The public Nostrian packages do not bundle preserved Esterian Conquest executables, manuals, or DOS helper assets.

2.4. Choose Your Deployment

Use one of these live paths today.

2.4.1. Localhost Session

Use this for direct same-machine play, hotseat testing, or a small trusted session.

1. Build the binaries from source.
2. Create a game directory with `nc-sysop new-game`.
3. Run `nc-game` directly.
4. Run `nc-sysop maint` when the year should advance.

The simplest localhost setup is:

```
nc-sysop new-game /home/sysop/nc-games/friday-night --name "Friday Night NC" --players 4
nc-game --dir /home/sysop/nc-games/friday-night --player 1
```

2.4.2. BBS Door Host

Use this when you want `nc-door` under Mystic, ENiGMA½, Synchronet, WWIV, or a similar BBS.

1. Create the game directory.
2. Write a per-game `config.kdl` with any fixed-seat reservations.
3. Initialize it with `nc-sysop new-game --bbs --players N`.
4. Launch `nc-door` with a BBS dropfile.
5. Keep maintenance outside the door. Run `nc-sysop maint` from the host or the BBS event runner.

For BBS hosting, use the packaged `nc-sysop` archive or build from source.

IMPORTANT: A planned Nostr/GameServer path exists, but it is not part of the live `nc-sysop` workflow today. For live games, use localhost or BBS hosting.

For the exact launcher setups, see:

- [docs/sysop/bbs/mystic-bbs-setup.md](#)
- [docs/sysop/bbs/enigma-bbs-setup.md](#)
- [docs/sysop/bbs/synchronet-bbs-setup.md](#)

- docs/sysop/bbs/wwiv-bbs-setup.md

2.5. Game Directory Layout

Direct nc-game campaigns are DB-only:

```
/path/to/mygame/
ncgame.db
```

BBS door campaigns keep a minimal live config file beside the runtime DB:

```
/path/to/mygame/
config.kdl
ncgame.db
```

All tools take `--dir /path/to/mygame` to find the game.

2.6. Initializing a New Game

Create a new game with `nc-sysop new-game`:

```
sudo -u ncgame nc-sysop new-game /srv/nc/games/friday-night --name "Friday Night NC" --
players 4
```

This creates one runtime file: `ncgame.db`.

For BBS door campaigns, write `config.kdl` first and then initialize with:

```
nc-sysop new-game --bbs --players 4 /path/to/mygame
```

Use these creation flags:

- **--name:** String. Optional game title stored in `ncgame.db`. If you omit it, `nc-sysop` builds a title from the directory slug.
- **--players:** Integer. Number of empires. Supported range: 1-25. Default: 4.
- **--seed:** Integer. Optional campaign RNG seed. It controls map layout, starting positions, and all random events. If you omit it, the engine picks a random seed and saves it to `ncgame.db`. The seed cannot be changed after creation.
- **--bbs:** Flag. Initialize a BBS campaign from an existing per-game `config.kdl`. In this mode `--name` is not accepted; put `game_name` in the file. Use `--seed` only when you want a reproducible map for that one campaign.

NOTE: Use a different seed for every game. Reusing the same seed produces the same map, starting positions, and event sequence every time. For BBS campaigns, keep that override on the `new-game --bbs --seed ...` command line instead of storing it in `config.kdl`.

The target directory basename becomes the stable game slug. It must use only lowercase ASCII letters, digits, and dashes. The slug is not the human-readable `game_name`.

3. Game Directory Structure

- `ncgame.db`: The SQLite runtime database. All live game state is here. Do not edit it by hand.
- `config.kdl`: The game's settings, which you edit. See Section 4.4.

4. Configuration

Keep the live paths separate:

4.1. Direct nc-game

- Live state: per-game `ncgame.db`
- Runtime path: `nc-game --dir ... --player ...` on localhost
- Operator rule: use the normal `nc-sysop settings ...` commands and schedule `nc-sysop maint yourself`

4.2. BBS Door Host

- Live state: per-game `config.kdl` plus `ncgame.db`
- Runtime path: `nc-door` with a dropfile; ENiGMA½ also passes `--socket-port` and Windows Synchronet passes `--socket-descriptor`
- Operator rule: write `config.kdl` before `nc-sysop new-game --bbs`
- Common hosts: Mystic, ENiGMA½, and Synchronet on Windows or Linux; WWIV on Linux with its `CHAIN.TXT` drop file

Use `nc-sysop settings ...` to inspect or change any campaign:

```
nc-sysop settings show --dir /srv/nc/games/friday-night
nc-sysop settings set --dir /srv/nc/games/friday-night --game-name "Friday Night NC"
nc-sysop settings reserve --dir /srv/nc/games/friday-night --player 1 --alias SYSOP
```

For the full non-BBS settings reference, including the raw `settings show` shape and the advanced carried-forward state fields, see `docs/sysop/rust/campaign-settings.md`.

4.3. AI Seats

A non-BBS campaign can seat the learned AI in place of a player:

```
nc-sysop settings ai-seat --dir /srv/nc/games/friday-night --player 3 --difficulty hard
nc-sysop settings ai-seat --dir /srv/nc/games/friday-night --player 4 --difficulty
normal --personality rusher
nc-sysop settings remove-ai-seat --dir /srv/nc/games/friday-night --player 3
```

`--personality` picks the seat's style. The default is `land-grab`, which expands hard and early and strikes when it has the odds. `rusher` goes to war at once and attacks at long odds. `turtle` keeps strong garrisons and starbases, always guards threatened worlds, and attacks late. Different seats can use different styles.

Difficulty is `easy`, `normal` or `hard`. `hard` is the agent at full strength. On test maps it won about two games in three against three scripted executor opponents. `normal` won about three in five, and `easy` about one in five.

A reserved seat cannot be an AI seat. Each maintenance year, every AI seat commissions its ships and submits its turn before the year runs, as a player would. It sees only what a player of that seat could see. Its memory is kept in `ncgame.db` between runs.

Maintenance needs the agent file. Set `NC_AI_MODEL` to the path of its `.onnx` file wherever `nc-sysop maint` runs. A campaign with AI seats refuses maintenance without it.

4.4. Game settings: config.kdl

Each game keeps its settings in `config.kdl`, in the game directory. It is the counterpart of classic ECUTIL's program options. `nc-sysop new-game` writes it with the defaults. For a BBS campaign you may write it first; `new-game --bbs` reads it, then writes it back in the form below.

```
// Set by nc-sysop new-game and fixed for the game; shown for reference only.
// players 4, seed 8211, map 18x18
game_name "Friday Night"
theme "tokyo_night"
snoop #true
session {
    max_idle_minutes 10
    minimum_time_minutes 0
    local_timeout #false
    remote_timeout #true
}
inactivity {
    autopilot_after_turns 3
}
maintenance {
    enabled #false
    interval_minutes 1440
}
reservations {
    seat player=1 alias="SYSOP"
}
ai_seat player=3 difficulty="normal" personality="land-grab"
```

Every field may be left out. A missing field takes the default shown above. `game_name` and `theme` left out keep what the game already has.

- The first two lines are a comment showing the number of empires, the seed and the map size. `new-game` sets them and they never change. Editing the comment does nothing. A `players`, `seed` or `map_size` field is refused with an error naming it, and logins and maintenance stop until you remove it.
- **snoop** and **session**: As described under Section 7.2.
- **inactivity.autopilot_after_turns**: Put an empire on autopilot after this many missed turns, 0 to 100. 0 turns it off. The default is 3. An empire whose player gave orders for 3000 and none since goes on autopilot in 3003's maintenance. You may change it during a game, for instance to fix a wrong value.
- **maintenance**: The schedule your timer reads. See Section 9.
- **reservations**: Seats held for named BBS callers. See Section 10.1.
- **ai_seat**: A seat the computer plays. Only an open seat can become an AI seat; one a person has joined is refused. See Section 4.3.

Edits take effect at the next login or maintenance run. Then every empire gets a “Game Notice” report listing what changed: the game name, the autopilot rule, session limits, the maintenance schedule and which empires the computer plays. The theme, snoop and reservations are not announced.

`nc-sysop settings set, reserve, unreserve, ai-seat` and `remove-ai-seat` edit this file for you. They rewrite it, so comments you added are lost.

4.5. Display Defaults

`default_theme_key` sets the default display look. That key names a compiled-in color set. It is not a file path.

Players may still choose a local color theme inside `nc-game`. That choice is stored in `ncgame.db` as a player preference.

In BBS door mode, `nc-door` does not use `default_theme_key` at runtime. Door sessions always force the bundled `mag16` palette so ANSI16 terminals and BBS clients get a predictable baseline.

5. Terminal Access

`nc-game` is a direct terminal client. The main supported path is local same-machine play. If you remote into a trusted shell and launch `nc-game` there, it behaves the same way. No special session flags are needed.

The client detects color mode from the environment:

- `COLORTERM=truecolor` → 24-bit RGB
- `TERM` containing `256color` → 256-color
- Otherwise → 16-color ANSI fallback

Force a mode with `--color-mode` if your terminal does not pass `COLORTERM` through reliably:

```
nc-game --dir /path/to/mygame --player 1 --color-mode 256
```

UTF-8 encoding (the default) is correct for normal terminals. Use `--encoding cp437` only when you are proxying through a BBS or a CP437 terminal emulator over SSH.

6. Localhost Session Setup

Use these steps for localhost play:

1. Create or choose a game directory.
2. Run `nc-game` directly in your terminal:

```
nc-game --dir /path/to/mygame --player 1
```

3. Run `nc-sysop maint /path/to/mygame` when you want the year to advance.

Color mode and encoding default to `auto / utf8`. The client reads `COLORTERM` and `TERM` on its own.

7. BBS Door Setup

`nc-door` is the standard BBS entrypoint on Windows and Linux. Use it when the host is a BBS. Keep `nc-game` for direct localhost sessions. In door mode, `nc-door` uses CP437 and 16-color ANSI for classic terminal clients.

7.1. Flags for Door Mode

```
nc-door --dir /path/to/mygame --dropfile /path/to/D00R32.SYS --encoding cp437 --color-mode ansi16
```

--encoding cp437 switches box-drawing and other extended characters to the CP437 code page. Classic ANSI BBS terminals (SyncTERM, NetRunner, etc.) need it to render the screen correctly.

When --encoding cp437 is active, --color-mode defaults to ansi16. Override it only if you know your BBS clients support a richer color depth.

Door sessions also force the bundled mag16 theme, whatever the campaign's default_theme_key says. The in-game **A>nsi color ON/OFF** command switches between that ANSI16 palette and a greyscale monochrome view.

7.2. Drop File

nc-door can read a BBS drop file directly with --dropfile <path>:

```
nc-door --dir /path/to/mygame --dropfile /path/to/D00R32.SYS
```

nc-door detects the format from the filename, ignoring case. Supported formats:

- D00R32.SYS — modern standard (Enigma, Mystic, Talisman, etc.)
- D00R.SYS — legacy, widest BBS software support
- CHAIN.TXT — WWIV format

The drop file supplies the player alias and session time limit. Explicit CLI flags always override drop file values. Drop files may be written in CP437 or UTF-8. When --dropfile is given and --encoding is not, encoding defaults to cp437.

D00R.SYS supplies the handle from line 36. When line 36 is missing or blank, it supplies the real name from line 10 instead. When the handle matches no seat, nc-door tries the real name. The real name resumes a seat reserved under that name, or a joined seat not yet bound by handle. That seat then takes the caller's handle. A seat is bound by handle once its player joins or enters it with a handle. From then on the real name never matches it. So a caller whose real name equals another player's handle cannot enter that player's empire.

A D00R.SYS with no handle on line 36 gives only the real name. When that name matches the handle of a seat bound by handle, nc-door refuses the login. The caller is told that the login could not be matched and that the sysop has been notified. nc-door writes the real name and the seat number to standard error, and to the log when --log-file is set, then exits with status 1. To fix it, set the BBS to write handles on line 36. If the caller owns that seat, you can instead reserve it under the real name:

```
nc-sysop settings reserve --dir /path/to/mygame --player 2 --alias "Real Name"
```

--timeout <minutes> sets a session time limit, with or without a drop file.

In door mode the time limit is enforced. When it runs out, nc-door shows a short message and exits normally, as if the caller had quit. The game saves state as each action is applied, so nothing is lost. When the limit is longer than two minutes, the caller sees a one-line warning

two minutes before it runs out, on the row above the command line. The next screen redraw clears it. The time limit counts from door launch.

Door mode also has an idle limit. It comes from the campaign setting `session_max_idle_minutes`, not from an `nc-door` flag. Set it with `nc-sysop settings set --dir /path/to/mygame --session-max-idle <minutes>`. The default is 10 minutes. The most is 120. 0 turns it off. Only a keypress counts as activity. The idle clock restarts at each key. When no key arrives in that time, `nc-door` shows “No key was pressed for too long. Returning to the BBS.” and exits normally.

Players of one campaign can be in sessions at the same time. Each change a player makes is applied to the latest saved state and saved in one database transaction. No player’s change undoes another’s, and each session then shows the others’ changes. The same holds for `nc-game submit-turn`.

Keep the game directory on a local disk of the machine that runs the doors and maintenance. Sessions share `ncgame.db` through SQLite’s write-ahead log. That works only between processes on one machine. A network share, mapped drive or synced folder can fail saves or damage the file. Beside `ncgame.db` you will see `ncgame.db-wal` and `ncgame.db-shm` while sessions are open. They belong to the database. Never delete them, and copy the game only while no session or maintenance is running.

A session that stays open across a maintenance run cannot save into the finished year. Its next save is refused, and the caller is told “Maintenance has run; your last change was not saved. Log in again.” In door mode `nc-door` shows that line and exits normally, as it does at the time limit. A local `nc-game` session shows the line and stays open without saving. `nc-game submit-turn` fails with the same message if maintenance runs between its load and its save.

Reserve seats in the BBS campaign’s `config.kdl` when you want the caller alias to pick the empire. Edit the file directly or use:

```
nc-sysop settings reserve --dir /path/to/mygame --player 1 --alias SYSOP
nc-sysop settings reserve --dir /path/to/mygame --player 2 --alias NightShade
```

With that in place, a reserved caller can launch with `--dropfile` alone. If the caller alias is not reserved, `nc-door` still uses the same dropfile-only door flow:

- if the alias already matches a stored joined player handle, that caller resumes the matching empire
- if the alias does not match an existing joined empire, the caller lands on the BBS first-time menu. He claims the lowest-numbered open unreserved empire only when he confirms the join
- if the game is full, the caller still reaches the BBS first-time menu, but J is refused with the normal no-open-empires notice

NOTE: The original DOS `ECGAME.EXE v1.5` expects a strict 32-line WWIV-style `CHAIN.TXT` drop file. Enigma BBS-generated `DOOR.SYS` / `DORINFO` files will crash `ECGAME.EXE`. See the

legacy DOS compatibility section in `docs/sysop/bbs/enigma-bbs-setup.md` if you need to host the original binary.

7.3. Verified BBS Hosts

The native Rust `nc-door` binary is verified on Mystic, ENiGMA^{1/2}, and Synchronet. WWIV has a validated Linux path. The core launch is the same on every host: pass `--dir` and a dropfile. ENiGMA^{1/2} also passes `--socket-port`, and Windows Synchronet also passes `--socket-descriptor`. The helper wrapper at `tools/bbs/run_nc_rust.sh` is for source-tree testing only. It is not the normal packaged door path.

Mystic uses the direct dropfile path:

```
nc-door --dir /path/to/mygame --dropfile /path/to/D00R32.SYS
```

ENiGMA^{1/2} uses the same dropfile plus its temporary localhost socket server:

```
nc-door --dir /path/to/mygame --dropfile /path/to/D00R32.SYS --socket-port {srvPort}
```

Windows Synchronet also passes the inherited socket descriptor:

```
nc-door --dir /path/to/mygame --dropfile %f --socket-descriptor %H
```

Linux Synchronet is validated with the same `D00R32` dropfile. If your Synchronet install mangles a long `cmd=` line, put the full `nc-door --dir ... --dropfile ...` command in a small wrapper script and pass only `%f` to that wrapper.

WWIV uses a validated Linux `CHAIN.TXT` path:

```
nc-door --dir /path/to/mygame --dropfile /path/to/CHAIN.TXT
```

Treat native Windows WWIV as unvalidated for now. In one Windows 11 test with a 2023 WWIV Windows package, Defender quarantined the downloaded binary, and `bbs.exe` then crashed before login on both remote and local nodes.

For the exact launcher setups, see `docs/sysop/bbs/mystic-bbs-setup.md`, `docs/sysop/bbs/enigma-bbs-setup.md`, `docs/sysop/bbs/synchronet-bbs-setup.md`, and `docs/sysop/bbs/wwiv-bbs-setup.md`.

8. File-Based Turn Submission

Use this when a player needs to submit a KDL turn file directly:

1. Validate the file without writing:

```
nc-game submit-turn --check --dir /path/to/mygame --player 1 --file /path/to/player1-turn.kdl
```

2. Apply it:

```
nc-game submit-turn --dir /path/to/mygame --player 1 --file /path/to/player1-turn.kdl
```

The `--player` value must match the `turn player=...` header in the file. If any order in the file is invalid, the whole submission is rejected and nothing is written. This command applies the file directly. It is not a queue or an upload inbox.

A minimal turn file:

```
turn player=1 year=3000
tax rate=37
```

A fuller example:

```
turn player=1 year=3000

tax rate=37

planet record=16 {
    build points=4 kind="scout"
}

fleet record=1 {
    order speed=3 kind="scout_system" x=16 y=13
}

message to=2 subject="Border" body="Watching the north lane."
```

A `seek_home` or `salvage` order may leave out `x` and `y`. The fleet then heads for the nearest world its player owns, the same world the game would pick.

`speed value=N` inside a `fleet` block changes only that fleet's speed, as the in-game speed change does.

For the full node reference and schema, see `docs/sysop/rust/turn-kdl.md`.

9. Turn Processing and Maintenance

Run yearly maintenance with:

```
nc-sysop maint /path/to/mygame [turns]
```

`nc-sysop maint` advances the campaign in `ncgame.db` at once. It does not change the schedule fields. NC does not schedule maintenance by itself.

While it runs, players can stay logged in, but their changes are refused with “Campaign paused; your last change was not saved. Try again later.” This keeps the year it resolves from changing under it. When it finishes, their sessions are told that maintenance has run and they must log in again. Only one maintenance run goes at a time for a campaign. A second is refused. If a run is killed or the machine stops, players stay paused until you run `nc-sysop maint` again. That run reports the interrupted run and runs the year.

Every session checks the campaign file when it opens it. A damaged file is reported as “the campaign file ... is damaged; tell the sysop” instead of being played. Restore it from a backup.

Run maintenance by one of these methods:

- a `systemd` timer
- `cron`
- a BBS event runner
- or manual `sysop` operation

A non-BBS campaign stores three schedule fields in `ncgame.db`. They are notes for you and your scheduler. Nothing in `nc-sysop` or `nc-game` reads them to start a run.

- **maintenance_enabled:** on or off. Default: off. Marks the game as scheduled.
- **maintenance_interval_minutes:** Minutes between runs. Default: 1440, one day. Must be at least 1.
- **maintenance_next_due_unix_seconds:** When the next run is due, as a Unix timestamp in seconds. Blank until you set it.

`nc-sysop settings show` prints all three. Set them with `settings set`:

```
nc-sysop settings set --dir /srv/nc/games/friday-night --maintenance-enabled on --
maintenance-interval-minutes 10080 --maintenance-next-due 1775347200
```

That example records a weekly schedule. 10080 is seven days in minutes. 1775347200 is only a sample Unix timestamp. Replace it with the first due time you want. Pass `--maintenance-next-due ""` to clear the due time.

`nc-sysop maint` never changes these fields. If your timer reads them, it must update the due time itself after each run. The turn runs when your timer or you run `nc-sysop maint`. `enabled` and `interval_minutes` are also in `config.kdl`; the due time is kept only in `ncgame.db`.

9.1. Checking a Campaign

To check that a campaign is safe to play on, use:

```
nc-sysop check --dir /path/to/mygame
```

It checks three things and stops at the first failure:

- **File:** `ncgame.db` is intact. A damaged file, from a bad disk or a copy made while the game was running, is reported as damaged.
- **Year:** the latest saved year loads.
- **Maintenance:** the campaign passes the checks `nc-sysop maint` makes before each turn. Any problem is listed.

It changes nothing. It exits with an error when a check fails, so a script can run it before maintenance or after restoring a backup.

10. Player Management

Each campaign tracks when its players last took part in `ncgame.db`. The inactivity autopilot threshold is set in `config.kdl` (see Section 4.4); the default is 3 turns and 0 turns it off. A player counts as active for the year when he reaches the live `nc-game` menus or applies a `submit-turn` file. When inactivity turns autopilot on, the player gets a report saying so.

A defeated player keeps one final review-only login. If the empire has no planets, the next maintenance turns it into a computer-controlled rogue empire. The former player's identity and final-review state stay in the store. This does not reopen the seat or restore orders. A planetless rogue patrols with its warships and does not compete for the throne.

10.1. Reserving Seats

To reserve empire slots for specific BBS users, edit the per-game BBS `config.kdl` or use:

```
nc-sysop settings reserve --dir /path/to/mygame --player 1 --alias SYSOP
nc-sysop settings reserve --dir /path/to/mygame --player 2 --alias NightShade
```

Each `seat` entry binds one 1-based empire slot to one caller alias. Alias matching is ASCII case-insensitive, so `NightShade`, `nightshade`, and `NIGHTSHADE` count as the same reservation.

With reservations in place, launch `nc-door` with `--dropfile` and let the caller alias choose the empire:

```
nc-door --dir /path/to/mygame --dropfile /path/to/D00R32.SYS
```

IMPORTANT:

- if the caller alias is reserved, `--dropfile` alone is enough
- if the caller alias is not reserved and already matches a stored joined player handle, that caller resumes the matching empire
- if the caller alias is not reserved and does not match an existing joined player handle, the caller starts on the BBS first-time menu
- if a reservation conflicts with a different stored player handle, `nc-door` stops with a clear error so the sysop can reconcile the reservation and the runtime state
- if no open unreserved empires remain, J from the BBS first-time menu is refused and the caller stays on that menu

Reserving a seat does not join or name the empire by itself. It only routes the caller to the intended slot. The first-time join flow still claims an empire only when the caller confirms the join. That first join records the caller alias for later dropfile logins.

For localhost and direct console play, use `nc-game` and follow the Localhost Session Setup section earlier in this manual instead of the BBS dropfile path.

10.2. Exporting a Player's Star Map

Players can export their own star maps from the game. To export one for a player, for a board's file area or a game report, use:

```
nc-sysop map-export --dir /path/to/mygame --player 2
```

It writes what that player knows of the galaxy this year, just as the in-game export does: `ECMAP-P2-Y<year>.TXT`, `.CSV`, and `-DETAILS.CSV` in `<game_dir>/exports`. Use `--out-dir <path>` to write them somewhere else.

Leave out `--player` to export the galaxy map instead:

```
nc-sysop map-export --dir /path/to/mygame
```

It marks every world with `*` and shows nothing any empire has learned: no names, owners, or strengths. This is the map every player starts with, so it is safe to post for all players. It writes `ECMAP-GALAXY-Y<year>.TXT` and `.CSV`.

10.3. Checking a Player's Economy

To see one empire's economy for the current year, use:

```
nc-sysop economy-report --dir /path/to/mygame --player 2
```

It prints the empire's tax rate, production, stored points, and ranks, then each planet it owns with its production, revenue, growth, build capacity, armies, and batteries. This is private to that player, so use it to settle questions, not to share.

11. Terminology

- **game directory:** The directory containing `ncgame.db` for one running game. Pass it to all tools with `--dir`.
- **nc-sysop:** The Rust command-line sysop tool for campaign creation, maintenance, and settings management.
- **nc-game:** The Rust TUI player client for direct localhost sessions.
- **nc-door:** The Rust BBS door entrypoint. It runs the same game flow as `nc-game`, but it is the binary shipped for Windows and Linux BBS hosts.
- **ncgame.db:** The SQLite database that is the runtime source of truth for the Rust engine.
- **non-BBS campaign settings:** The sysop-managed runtime policy rows stored in `ncgame.db` for direct `nc-game` campaigns. They control game name, the default compiled-in color set, maintenance metadata, optional alias reservations, and a small set of carried-forward classic setup fields.
- **config.kdl:** The game's settings file. See Section 4.4.

12. CLI Reference

12.1. nc-sysop

```
nc-sysop [--log-file <path>] [--log-level <level>] <subcommand> [options]
```

Logging flags come before the subcommand:

- **--log-file <path>:** Append diagnostic log lines to this text file. Off by default. Missing parent folders are created.
- **--log-level <error|warn|info|debug|trace>:** How much to log. Default: `info`.
- **new-game:** Create a new campaign directory. Direct `nc-game` campaigns use `--name`, `--players`, and `--seed`. BBS campaigns use `new-game --bbs` with `--players` and `--seed`, and read `game_name` and the other settings from a `config.kdl` you write first. Every new game gets a `config.kdl`.
- **settings show|set|reserve|unreserve:** Inspect or edit the game's settings. Changes are written to `config.kdl` and `ncgame.db`.
- **maint:** Run one or more maintenance turns against `ncgame.db`.

- **map-export --dir <game_dir> [--player <N>] [--out-dir <path>]:** Write one player's star map as ECMAP-P<N>-Y<year> .TXT, .CSV, and -DETAILS.CSV files. Without --player, write the galaxy map with no empire's intel as ECMAP-GALAXY-Y<year> .TXT and .CSV. Default folder: <game_dir>/exports.
- **economy-report --dir <game_dir> --player <N>:** Print one empire's economy for the current year: totals, ranks, and each planet it owns.
- **check --dir <game_dir>:** Check that ncgame.db is intact, its latest year loads, and maintenance would accept it. Changes nothing.

12.2. nc-game and nc-door

```
nc-game --dir <game_dir> [--player <1-based index>] [options]
nc-door --dir <game_dir> --dropfile <path> [options]
nc-game submit-turn [--check] --dir <game_dir> --player <record> --file <turn.kdl>
```

Use nc-game for direct localhost sessions. Use nc-door for BBS door launches. The interactive flags below apply to both unless a host guide says otherwise.

Interactive client flags:

- **--dir <path>:** Game directory containing ncgame.db. Required.
- **--player <N>:** 1-based empire index for direct localhost play. Normal BBS dropfile launches can omit it.
- **--encoding <utf8|cp437>:** Output encoding. Default: utf8. Use cp437 for BBS or door mode.
- **--color-mode <ansi16|256|truecolor|auto>:** Color depth. Default: auto from the environment. CP437 mode defaults to ansi16.
- **--dropfile <path>:** Parse a BBS drop file (D00R32.SYS, D00R.SYS, or CHAIN.TXT). It supplies the alias and timeout, defaults encoding to cp437, and resolves the player seat through BBS config.kdl reservations or stored joined-player aliases.
- **--socket-port <value>:** Connect back to a localhost door socket. Mainly for ENiGMA½ abracadabra in socket mode.
- **--socket-descriptor <value>:** Native Windows door socket handle. Mainly for Synchronet-style socket door launches.
- **--timeout <minutes>:** Session time limit in minutes. It overrides any drop file value. Door mode ends the session when it runs out.
- **--export-root <path>:** Folder where the star map export writes its ECMAP-P<player>-Y<year> .TXT, .CSV, and -DETAILS.CSV files. Default: <game_dir>/exports. Also set by NC_GAME_EXPORT_ROOT.
- **--queue-dir <path>:** Optional second folder. When set, each star map export is also copied there, for a BBS download or queue area. Off by default. Also set by NC_GAME_QUEUE_DIR.
- **--log-file <path>:** Append diagnostic log lines to this text file. Off by default. Also set by NC_GAME_LOG_FILE.
- **--log-level <error|warn|info|debug|trace>:** How much to log. Default: info.

submit-turn flags:

- **--check:** Validate the KDL file without changing the campaign.
- **--dir <path>:** Game directory containing ncgame.db. Required.

- **--player <N>:** 1-based empire index. Required, and it must match the KDL header.
- **--file <path>:** Turn submission KDL file to validate or apply. Required.

NOTE: `nc-game submit-turn` is all-or-nothing. If any command in the file is invalid, the entire submission is rejected and nothing is written to `ncgame.db`.