NFI quick usage manual

`nfi` displays information about Nintendo Switch files in NSP, NSZ, XCI, XCZ, and NRO formats. It can also verify integrity, generate canonical names, and rename files.

1. Configure keys

NFI does not include keys. Use your own `prod.keys` file.

The simplest option on Linux is to place it at:

```
~/.switch/prod.keys
```

You can also specify the file or directory with `-k`:

```
./bin/nfi -k ~/.switch/prod.keys game.nsp
./bin/nfi -k ~/.switch game.nsp
```

On Windows:

```
nfi.exe -k C:\Users\user\.switch game.nsp
```

2. Display information

Analyze a file:

```
./bin/nfi game.nsp
Title ID: 0100000000000000
Base Title ID: 0100000000000000
Title Name: Game Name
Display Version: 1.1.1
Version: 0
System Version: 18.0.0
Masterkey: 16
Title Key: 00000000000000000000000000000000
...
Type: Base
Distribution: Digital
Structure: CDN
Signature: Passed
Permission: Safe
Error:
```

Paths containing spaces must be quoted:

```
./bin/nfi "Game Name [0100000000000000][v0].nsz"
```

Analyze all files in a directory:

```
./bin/nfi /games/
```

Analyze multiple paths:

```
./bin/nfi game.nsp update.nsz /games/
```

3. Generate a canonical name

To print only the canonical name:

```
./bin/nfi --canonical game.nsp
Game Name [0100000000000000][v0].nsp
```

This command does not rename the file.

4. Rename a file

Use `-r` or `--rename`:

```
./bin/nfi -r game.nsp
```

NFI displays the new name and does not overwrite an existing file. To preview the result, run `--canonical` first.

5. Verify integrity

Basic verification:

```
./bin/nfi --verify game.nsz
```

will generate an output like:

```
Integrity: Valid
Authenticity: Authentic
Installability: Ready
Container structure: OK
NCA filesystem headers: OK (5/5)
CNMT manifest: OK
NCA internal hashes: OK (4/4)
NCA content hashes: OK (4/4)
Rights and title key: OK
```

Verification with a progress bar:

```
./bin/nfi --verify=verbose game.xcz
[==              ]  14%  file 1/1 game.xcz content 1/4  NCA SHA-256  344.0 MiB/2.3 GiB
```

`--verify=1` has the same effect as `--verify=verbose`.

You can also verify an entire directory:

```
./bin/nfi --verify=verbose /games/
```

By default, up to four files remain active. To select a different value:

```
./bin/nfi --verify=verbose --parallel=8 /games/
```

Heavy work uses at most half of the detected CPUs. `GOMAXPROCS` reduces this limit when necessary:

```
GOMAXPROCS=2 ./bin/nfi --verify --parallel=4 /games/
```

The result reports:

- `Integrity`: file integrity;
- `Authenticity`: NCA authenticity;
- `Installability`: installation readiness.

6. Export results

Export verification results to CSV:

```
./bin/nfi --verify --export verification.csv /games/
```

Export to Excel with progress:

```
./bin/nfi --verify=verbose --export verification.xlsx /games/
```

Export regular game information:

```
./bin/nfi --export games.csv /games/
```

7. Main options

Option	Use
`-h`, `--help`	Displays help.
`-v`, `--version`	Displays version, commit, and build date.
`-k`, `--keys`	Specifies `prod.keys` or its directory.
`--canonical`	Prints only the canonical name.
`-r`, `--rename`	Renames to the canonical name.
`--verify`	Verifies integrity.
`--verify=verbose`	Verifies while displaying progress.
`--parallel=N`	Keeps up to `N` files under verification; default: 4.
`-x`, `--export`	Exports to CSV or XLSX.
`-s`, `--sort`	Sorts by `filename`, `titleid`, or `titlename`.

8. Help

To display every option:

```
./bin/nfi -h
```

If the binary is installed in `PATH`, omit `./bin/` and use only `nfi`.